Data-based Wave Filtering Using Neural Networks with Application to Dynamic Positioning

Andreas Franz

Master's thesis in Process and Systems Engineering
Supervisors: Jonas Linder & Jari Böling
Process Control Laboratory
Faculty of Science and Engineering
Åbo Akademi University
September 2018

Abstract

Ships are constantly affected by environmental disturbances. Oscillating wave forces are one of these disturbances. In dynamic positioning applications, these wave disturbances lead to an oscillating motion of the ship about its mean position. Even though the overall position is not affected by the oscillating disturbance, the actuators of the ship will try to compensate for it. This behavior is unwanted since it leads to unnecessary wear of the actuators and high fuel consumption. A solution is to apply wave filtering techniques, which eliminate the oscillating motion induced by waves from the measured position. This enables control of the low-frequency motion and reduces oscillation of the actuators.

This study evaluates if it is possible to learn a wave filtering algorithm to predict the low-frequency motion without knowledge about system parameters. For this approach, available measurement and control data is utilized to extract the information needed. A predictor based on a recurrent neural network is proposed. The recurrent neural network is trained on a sliding window to learn the model adaptively. The required training data is created from past measured position data. Creating the training data is done by estimating the dominating frequency of the wave disturbance and applying noncausal filtering. The recurrent neural network learns to predict the low-frequency motion using measured positions, control inputs of the ship, and past predicted low-frequency positions as inputs to the neural network. After training, the low-frequency position is predicted by the predictor until the neural network is trained again. The training is performed regularly to enable convergence towards a wave-filtering model and to adapt to changing disturbances.

The performance is evaluated through simulations with increasing complexity. The available data can contains enough information to learn wave filtering. Simulations on a system with one degree of freedom indicate that a wave-filtering model can be learned if the system is excited. Increasing the complexity to three degrees of freedom shows that the predictor can be extended to predict the low-frequency motion of more complex systems. However, the predictor struggles to learn rotational dependencies. Initial results from applying the predictor on data from a ship simulator are promising but need further work.

Acknowledgments

This study is a master's thesis for the program Process and Systems Engineering at Åbo Akademi University. It was performed at ABB Corporate Research Center (CRC) in Västerås, Sweden. I had a great time carrying out this project and I want to thank the people responsible.

Thank you Jonas Linder, ABB, for being a dedicated supervisor, for giving me advice and structure, and keeping me on the right track.

Thank you Jari Böling, Åbo Akademi, for helping me find this interesting project and supporting me from a distance.

Thank you Shiva Sander-Tavallaey, ABB for giving me this opportunity.

Furthermore, thank you Lena for your support and proofreading, and last but not least, thanks to my fellow thesis workers at ABB for a great time in Västerås.

Contents

List of Abbreviations

Abbreviation	
DP	Dynamic Positioning
LF	Low Frequency
WF	Wave Frequency
KF	Kalman Filter
NN	Neural Network
MSE	Mean Squared Error
MSEREG	Means Squared Error with Regularization
FNN	Feedforward Neural Network
NNFIR	Neural Network Finite Impulse Response
NNARX	Neural Network AutoRegressive with eXogenous inputs
RNN	Recurrent Neural Network
NNOE	Neural Network Output Error
NNARMAX	Neural Network AutoRegressive Moving Average with eXogenous inputs
NNBJ	Neural Network Box-Jenkins
DOF	Degree Of Freedom
FFT	Fast Fourier Transform
STD	Standard Deviation
RMSE	Root Mean Squared Error
MRMSE	The Mean of Root Mean Squared Errors

List of Notations

Neural Networks	
w_j	Weights between input layer and neuron j
w_{oj}	Bias of neuron j
v_j	Weights between neuron j and output layer
v_0	Biases of output layer
y_j	Output of neuron j
y	Output vector of NN
x	Input vector
n_i	Number of inputs
n_o	Number of outputs
n_h	Number of hidden nodes
$\Phi(\cdot)$	Nonlinear activation function
y_T	Target
ε	Error
$\boldsymbol{J}$	Jacobian matrix
μ	Adaptive parameter for the Levenberg-Marquardt algorithm

Vessel Dynamics	
$\boldsymbol{\eta}$	Generalized position vector
n	North position
e	East position
d	Down position
ϕ	Roll angle
θ	Pitch angle
ψ	Heading angle
$\boldsymbol{\nu}$	Generalized velocity vector
u	Velocity in surge
v	Velocity in sway
w	Velocity in heave
p	Roll rate
q	Pitch rate
r	Yaw rate
$\boldsymbol{R}(\phi,\theta,\psi)$	Linear-velocity transformation matrix
$\boldsymbol{T}(\phi,\theta)$	Angular-velocity transformation matrix
$\boldsymbol{J}(\phi,\theta,\psi)$	Kinematic transformation matrix
$\boldsymbol{\tau}_{\text{control}}$	Forces from the actuator
$\boldsymbol{\tau}_{\text{wind}}$	Wind induced forces
$\boldsymbol{\tau}_{\text{waves}}$	Wave induced forces
$\boldsymbol{\nu}_r$	Relative velocity
$\boldsymbol{\nu}_c$	Velocity of the current
$\boldsymbol{C}(\boldsymbol{\nu}_r)$	Coriolis-centripetal matrix for the rigid-body and added mass
$\boldsymbol{D}(\boldsymbol{\nu}_r)$	Damping forces

$\boldsymbol{\tau}_{\text{wave}}^{\text{lin}}$	Linear wave disturbance forces
$\boldsymbol{\tau}_{\text{wave}}^{\text{nlin}}$	Nonlinear wave disturbance forces
ω_0	Dominating wave frequency
λ	Damping coefficient
σ	Wave intensity

Predictor	
$\mathcal{Z}^M$	Set of available data
M	Number of collected samples
$u(t)$	Control signal
$y(t)$	Measured generalized position
k	Number of control signals
l	Number of measured positions
$\mathcal{Z}_{TR,t}^N$	Training data set
N	Number of samples used for training
$y_T(t)$	Target data
Δ	Time between two neural network training sessions.
$y_{\text{est}}(t)$	Frequency estimation data set
γ	Number of samples used for frequency estimation
ω_{HP}	Cutoff frequency of high pass filter
α	Number of samples used for filtering
$H_{\text{f}}(q)$	Filter
ω_{notch}	Notch frequency of notch filter
ω_{LP}	Cutoff frequency of low-pass filter
$y_{\text{f}}(t)$	Filtered data
ϵ	Transient length
β	Appending length
$y_{\text{LF}}(t)$	Low-frequency position
$\hat{y}_{\text{LF}}(t)$	Estimated low-frequency position
$u_{\text{p}}(t)$	Data set of control signals used for prediction
$y_{\text{p}}(t)$	Data set of measured positions used for prediction
$\hat{y}_{LF,\text{p}}(t)$	Data set of low-frequency position estimate used for prediction
n_{u}	Number of delayed control inputs
n_{y}	Number of delayed measured positions
n_{fb}	Number of delayed feedback values

1 Introduction

This chapter provides the reader with the background for this thesis. It presents the main reasons for this project and the goals. This master's thesis was carried out at ABB Corporate Research Center for the Control, Optimization and Analytics (COPAN) team.

1.1 Background

Automatic control of vessels has been around in the naval industry ever since Nicholas Minorsky studied the problem of automatic steering of ships back in 1922 (Bennett, 1984). Minorsky observed the manual control of a helmsman keeping the desired course to develop control theories. Since 1922, a lot has been achieved in the control of vessels. Heading autopilots and dynamic positioning are examples of applications. Dynamic positioning (DP) is part of the control of the horizontal motion of a ship by thrusters or propellers. The horizontal motion includes the motion of a vessel in surge, sway, and yaw (Fossen, 2011). A common factor for numerous control systems is that the controller has to get accurate information through feedback to achieve efficient control. The problem can be that the information required is not measurable directly, as in the case for dynamic positioning, where it is difficult to measure the linear velocities. The position measured, usually the position and the heading measured by GPS and gyrocompass, respectively, contains disturbances that are unwanted. These disturbances are especially oscillatory disturbances induced by waves. The traditional solution to this problem is called wave filtering (Fossen and Perez, 2009).

The disturbing forces affecting the ship can conceptually be divided into low-frequency (LF) and wave-frequency (WF) disturbances. The LF disturbances are forces like wind, current and wave induced drift, which have to be compensated for by the controller. The WF disturbances are oscillating forces with frequencies mostly outside the bandwidth of the vessel but inside the bandwidth of the actuators. These disturbances have a mean close to zero and do not need compensation in most DP applications (Fossen, 2011). To avoid the actuators compensating for every single wave, which will cause increasing wear and tear on the actuators and increasing fuel consumption, wave filtering can be applied to separate the LF and WF motion. Then, the controller is only provided with the LF information (Fossen, 2011).

Several techniques have been used to separate the LF motion from the WF motion. These are model-based observers or notch and low-pass filters (Lauvdal and Fossen, 1995). The problem with the filters is that they induce phase lag to the measurements, which affects the control performance. That is the reason why the model-based observers are used to a great extent (Fossen and Perez, 2009). The Kalman filter (KF) (Kalman, 1960) is widely applied for this problem. Later, a nonlinear passive observer (Fossen and Strand, 1999) was introduced, which compared to the KF, was simpler to tune, coupled with the fact that stability

of the overall system could be proven (Fossen, 2011). The main problem with the observer-based approach is that the performance is highly dependent on an accurate model of the vessel and the wave disturbance. Furthermore, information about the WF motion, like the peak wave frequency, is typically required (Fossen, 2011).

To overcome these obstacles, one possibility is to use machine-learning techniques. This allows the observer to learn how to separate the LF and WF motion with as little knowledge about the system as possible.

1.2 Purpose

The purpose of this study is to find ways to predict the LF position without requiring a lot of information about the system or accurate system and disturbance parameters. The knowledge gained from the results can later be used to develop other non-model-based observers with machine learning or improve existing model-based observers.

1.3 Project Goals

The overall goal of this study is to create a predictor, which is able to estimate the generalized LF position. The predictor should be able to operate without knowledge about the system and disturbances, and estimate the LF position without significant time delay.

1.3.1 Subgoals

1. Create a simplified model of the vessel and create appropriate training data.
2. Create a machine-learning-based algorithm that is able to predict the LF position.
3. Evaluate the algorithm on a more complex model.
4. Compare the results of the algorithm to a model-based nonlinear observer.

1.4 Thesis Limitations

The work of this study will be carried out within the following boundaries:

- Neural networks will be used as the machine-learning technique.
- The focus will be on separating the LF motion from the WF motion with a neural network, not to find an optimal neural network or method for doing so.
- Only low-speed applications will be considered.

2 Introduction to Neural Networks

This chapter gives a general introduction of the machine-learning technique used in this study and a brief overview of how neural networks (NN) have been used for problems similar to the one presented in Chapter 1. The first section is an introduction to machine learning. The following sections present the difficulties, advantages and structures of neural networks.

2.1 Machine Learning and Neural Networks

In a time where artificial intelligence is getting more and more attractive for businesses and applications worldwide, popular phrases like machine learning, NNs and the popular phrase deep learning can sometimes wrongly be used interchangeably. It is, however, true that these areas are related. Artificial intelligence is a general term and refers to the ability of computers to perform tasks that can be associated with human intelligence (Copeland, 2018). Machine learning can be seen as a kind of artificial intelligence and NNs as a machine-learning technique. Hence, applications using NNs fulfill the definition of machine-learning applications (Kim, 2017). Deep learning, which will not be discussed in this thesis, is also a machine learning technique that uses a more complex NN.

In brief, machine learning is a method where data is used to learn a model that describes the data (Kim, 2017). NNs are one such model that can be constructed. The learning part of machine learning is that the algorithm adapts the model itself from random initialized parameters. A learning technique called supervised learning is a popular technique for NNs (Kim, 2017). In supervised learning, an input sequence and a corresponding output sequence, the target data, is prepared by a teacher or supervisor. The prepared data is then used by the NN to learn the model. This is done by adjusting the parameters of the NN until the input can be mapped to the correct output. The learning procedure is often referred to as training and the data used is called training data. Supervised learning can be categorized into classification and regression. Examples for applications can be dynamic modeling, speech and image recognition, time series prediction, and many more. The main difference between modeling with machine learning and the classic first principle modeling approach is that machine-learning techniques can learn a model even though physical laws can be very complex or logic is involved (Kim, 2017).

2.2 Neural Network Basics

The smallest unit of a NN is called a neuron (Mandic and Chambers, 2001). A neuron usually consists of a set of parameters connected through a summation block to an activation function. These parameters, called weights, are gains of

their corresponding input. The mathematical description of one neuron is

$$y_j = \Phi(w_j^T x + w_{0j}), \tag{2.1}$$

where $\Phi(\cdot)$ is a nonlinear activation function, $x \in \mathbb{R}^{n_i}$ is the input, $y_j \in \mathbb{R}$ is the output, $w_j \in \mathbb{R}^{n_i}$ are the weights, and w_{0j} is the bias of the neuron j. The number of inputs is denoted n_i. By interconnecting a set of these neurons a NN can be built (Mandic and Chambers, 2001). The mathematical description of a NN with one layer of hidden neurons and a linear output layer is

$$y = \sum_{j=1}^{n_h} v_j \Phi(w_j^T x + w_{0j}) + v_0, \tag{2.2}$$

where $v_j \in \mathbb{R}^{n_o}$ is the weights from neuron j to the output layer, $v_0 \in \mathbb{R}^{n_o}$ are the biases of the output layer, and n_h is the number of neurons in the hidden layer (Cybenko, 1989). The number of outputs is denoted n_o. A graphic representation of (2.2) can be seen in Figure 2.1. The figure illustrates how the NN consists of three layers. The first layer, i.e. the input layer, contains the inputs to the NN. The second layer, i.e. the hidden layer, consists of the building blocks summation and the nonlinear activation functions. The third layer, i.e. the output layer, consists of a summation block and outputs the result of the NN. The nodes represented by a constant value of one are the biases of the nodes.

2.2.1 Activation Functions

The reason why NNs are good nonlinear function approximators is that zero-memory nonlinear functions are build into the NNs (Mandic and Chambers, 2001). These nonlinear functions are called activation functions. They introduce nonlinearity into the NN. Hence, a NN with only linear activation functions are not able to approximate nonlinear functions well. Together with a summation block, the activation function forms the neuron, see Figure 2.1. There exists a rich variety of nonlinear activation functions, of which some, as the threshold and the sigmoid function, are discussed in Haykin (2009). The sigmoid activation function is characterized by its "S"-shape. Examples of sigmoid activation functions are the logistic function

$$\Phi(x) = \frac{1}{1+e^{-x}} \tag{2.3}$$

with an output range from 0 to +1 and the hyperbolic tangent function

$$\Phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.4}$$

with an output range from -1 to +1. Different saturation values of the activation function can also easily be achieved by modifying the sigmoid functions. The sigmoid function is the most commonly used activation function in NN (Haykin, 2009). Depending on the application of the NN, an activation function can be applied to the output layer to scale the output into a certain output range. Cybenko

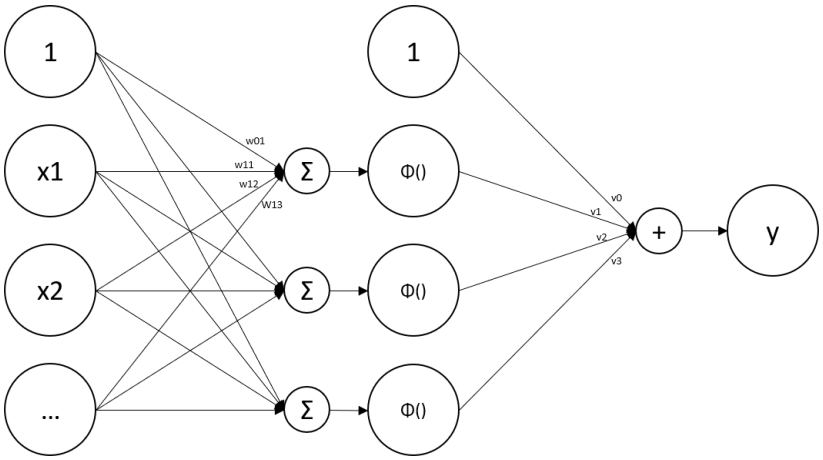


Figure 2.1: Schematic of a fully connected feedforward neural network

(1989) showed that a neural network with a single hidden layer and an appropriate number of sigmoid activation functions is able to approximate any arbitrary continuous function. An important requirement of the activation function is the continuous differentiability. This feature is important for gradient-based learning algorithms and learning algorithms that require the existence of a Hessian matrix or higher-order derivatives (Mandic and Chambers, 2001).

2.2.2 Neural Network Training

When training NNs, the goal is to map an input to the desired output. During training the error

$$\varepsilon_i = y_T(i) - y(i) \tag{2.5}$$

is calculated for every target-output pair i during training. The error is used in a loss function, which is non-negative and decreases with the quality of learning (Mandic and Chambers, 2001). Training the NN will minimize this function. A commonly implemented loss function is the mean squared error (MSE), which is defined as

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\varepsilon_i^2. \tag{2.6}$$

The loss function can be modified by adding penalty terms, for example, penalizing the size of the network, the size of the weights or the change of weights. This function, which is minimized during NN training, is also called objective function.

The backpropagation and the Levenberg-Marquardt algorithm are the most widely used training algorithms. The Levenberg-Marquardt algorithm was proven to be effective for training NN (Dias et al., 2006). The algorithm updates the weights according to

$$w_{t+1} = w_t - [\boldsymbol{J}^T\boldsymbol{J} + \mu\boldsymbol{I}]^{-1}\boldsymbol{J}^T\boldsymbol{\varepsilon}, \tag{2.7}$$

where w are all weights and biases, $\boldsymbol{J}$ is the Jacobian matrix with derivatives of the errors of the networks with respect to the weights and biases, and μ is

an adaptive parameter that shifts the algorithm between Newtons method and steepest descent. The vector of network errors is denoted $\boldsymbol{\varepsilon}$. The Jacobian can be calculated with a modification of a standard backpropagation technique (Hagan and Menhaj, 1994).

NN training can be either batch (offline training) or incremental (online training) (Mandic and Chambers, 2001). Apart from the application area, the difference between these methods is the way the data is presented to the NN during training. For batch training, the total error of the training data is calculated by the loss function. The weights are then updated according to the total error. The total error of the batch is calculated and the weights are updated until a stopping criterion is reached. During incremental training, on the other hand, the error is calculated and the weights are updated immediately after one input vector and the corresponding output are presented to the NN. This is repeated for all input-output pairs of the training data until a stopping criterion is reached (Mandic and Chambers, 2001).

A NN, which has been trained once and is used in an application without changing the weights further, has been subject to nonadaptive training. Contrary, adaptive training refers to the training of a NN during usage (Mandic and Chambers, 2001). Both adaptive and non-adaptive training can deploy batch or incremental training, for instance see Dias et al. (2006) for adaptive batch training and Abdollahi et al. (2006) for an incremental approach. The advantage of adaptive training is that past information can be forgotten and the NN can learn changing dynamics or processes and is suitable for applications with non-stationary behavior (Mandic and Chambers, 2001).

2.2.3 Generalization

Overfitting is a known challenge in all regression problems with noisy data. In system identification with the least squares method, the number of model parameters defines the order of the model. A high order model can be used to describe functions with high complexity. The data, which has to be approximated by the model, can be corrupted by noise. A high-order model could learn the noise in the data. This, in turn, will result in a very low error between the training data and the model output. However, describing any other data than the data trained on can be a challenge. Accordingly, overfitting is a state where the neural network is very good at describing the training data but is not able to describe new data. To avoid this, generalization has to be improved (Kim, 2017). There are two ways to improve generalization: to reduce the number of parameters or to reduce the effective range of the parameters. Early stopping and regularization are examples for reducing the effective range (Prechelt, 1998).

Early stopping is a method where the training data is divided into training and validation data. The validation data is not used for training but merely to prevent overfitting. This is done by continuously checking the performance of the NN on the validation data. Usually the fit on the validation data decreases when the network starts overfitting to the training data (Morgan and Bourlard, 1990).

Regularization is a generalization method which modifies the objective function to keep the weights small (Krogh and Hertz, 1992). The MSE loss function is

extended to a mean squared error with regularization (MSEREG) as

$$\text{MSEREG} = a\frac{1}{N}\sum_{i=1}^{N}\varepsilon_i^2 + b\sum_{j=1}^{n} w_j^2, \tag{2.8}$$

where a and b are hyper parameters. Foresee and Hagan (1997) propose the Bayesian regularization method, which can be used to find the optimal parameters a and b. Keeping the weights small results in a smoother response and can decrease the risk of overfitting (Foresee and Hagan, 1997).

2.2.4 Preprocessing

When updating the weights with a gradient-based training method, the weight updates are proportional to the value of the gradient and the input data. In order to perform efficient prediction, the range of the input data should be matched with the range of the chosen activation function together with the mean and variance (Mandic and Chambers, 2001). For example, the range of the hyperbolic tangent function is -1 to 1 with a mean of 0, which implies that the input data should be changed accordingly. Mandic and Chambers (2001) present examples for preprocessing methods :

- Normalization
  - Each element of the input vector is divided by the squared norm of the vector.
- Rescaling
  - The input vector is divided by a constant and a constant is added or subtracted.
- Standardization
  - Standardize the input vector to zero mean or another midrange and to a standard deviation or into a certain range.

2.3 Neural Network Structures

NNs can have a variety of different structures. The structures most commonly used to model dynamic systems are feedforward neural networks (FNNs) and recurrent neural networks (RNNs). The distinct difference between them are the feedback loops, which occur in RNNs but not in FNNs. Both FFNs and RNNs can act as models of dynamic system if the input vector is chosen to be a time delayed sequence. Dependent on which regressors are chosen as inputs, different models of dynamic systems can be represented by the NN. For a deeper insight on how NNs can be used for system identification and modeling of dynamic systems, see Narendra and Parthasarathy (1992) or Sjöberg et al. (1994).

FNNs are one option in modeling dynamic systems. Sjöberg et al. (1994) introduced the nomenclature to be able to distinguish different models of dynamic

system. They follow the nomenclature for linear models in system identification. For FNNs the finite impulse response model (NNFIR) and the autoregressive model with exogenous inputs (NNARX) apply. The NNFIR model uses only control inputs as regressor, while the NNARX model also uses delayed measured outputs.

The output of FNN are not dependent on the previous outputs of the NN. If the output of the NN depends on previous calculated values, one talks about RNN. Characterizing for a RNN is the feedback of delayed past outputs of the RNN or delayed internal states of the hidden layers (Sjöberg et al., 1994). Siegelmann et al. (1997) shows that a RNN using only delayed outputs is sufficient to approximate dynamic systems.

The feedback introduces memory to the RNN (Mandic and Chambers, 2001), which thereby gains increased flexibility and makes ths RNN a good approximator of dynamic systems. Therefore, they can be used in system identification (Narendra and Parthasarathy, 1992). Like in traditional system identification, different regressors can be used, which lead to different models. To describe the RNN model for system identification, Sjöberg et al. (1994) suggests the nomenclature NNOE model, NNARMAX model, and NNBJ model for the output-error model, the autoregressive moving average model with exogenous inputs, and a Box-Jenkins model, respectively. The input vector of the NNOE-model consists of the control input and delayed outputs of the RNN. The input vector of the NNARMAX model consists of the control input, delayed predicted outputs of the RNN, and past measured outputs. The NNBJ model uses all four different regressors as inputs. Using past measured outputs as regressor brings in past disturbances into the model, which enables the NN to model the disturbance, too. (Sjöberg et al., 1994).

2.4 Neural Networks and Dynamic Modeling

Neural networks have gained interest in the system identification community. Sjöberg et al. (1994) shows that NN can be used for system identification of nonlinear systems. The authors write about NN from the view of control engineering and present to choose suitable regressors for system identification.

Even though the fast Levenberg-Marquardt training algorithm is mostly used for offline training, Dias et al. (2006) proposes a method which uses the algorithm to identify a real system with a NN in a sliding window approach. The identified system is used to tune a controller. The authors also present a way to prevent overfitting by applying early stopping.

NNs have also been used to estimate states in different applications. Aydogmus and Aydogmus (2015) presents an algorithm to estimate the shaft speed of a dc-motor using NNs. The results are compared with an extended KF-based algorithm. A NN, trained offline on collected shaft speed data, is used to predict the speed of the dc-motor. The NN offline training is done with the Levenberg-Marquardt method. The results show a low error of the prediction.

Abdollahi et al. (2006) created a neuro-observer which uses a conventional observer to estimate states of a system and a neural network to identify the nonlinearity. The learning rule is modified to ensure robustness of the observer. The

proposed observer is demonstrated on simulations of flexible joint manipulators.

2.5 Challenges in Using Neural Networks

Almost all problems that occur when using NNs originates from the fact that the training data and the data used after training are distinct (Kim, 2017). This means that the NN probably will be used on data which it has not previously trained on. Typically NNs are good at interpolating but bad at extrapolating with respect to the training data (Lohninger, 1999). In other words, the training data should represent the whole range of data that the NN could come across.

3 Selected Topics in Ship Modeling and Estimation

An introduction to the dynamics of vessels and environmental disturbances will help the understanding of the choices made in this thesis. The ship model presented in this chapter will be used for simulation purposes and guide the choices of the simplified models presented in Chapter 5. Furthermore, these models also motivate the choices made in the observer algorithm presented in Chapter 4. Therefore, this chapter shortly presents the fundamentals of the vessel and disturbance dynamics and later gives an overview about past and current wave filtering techniques.

3.1 Vessel Kinematics

A vessel moves in six degrees of freedom (DOFs). In Figure 3.1 one can see how the motion of the vessel is defined. Surge is the forward motion of the ship and sway is the sideway motion of the ship. The yaw rate is the rotation of the ship. These three motions define the motion in the horizontal plane. Vessel models derived from these three DOFs are used in dynamic positioning applications, path-following systems, and trajectory-tracking control systems. Heading autopilots and forward speed controllers are designed on the basis of one DOF models with yaw and surge, respectively, as the chosen DOF. The remaining DOFs; roll, pitch, and heave are used in, for example, roll-damping or unmanned underwater vehicle control. Hence, these DOFs are presented to describe a full model, but are otherwise beyond the scope of this study. The interested reader is referred to Perez (2005).

It is convenient to describe the position and the motion of vessels with two reference frames, a local geographical Earth-fixed frame and a body-fixed frame (Perez, 2005). The local Earth-fixed frame has an origin located on the mean free-surface of the water. For a full six DOFs model the the position of a vessel

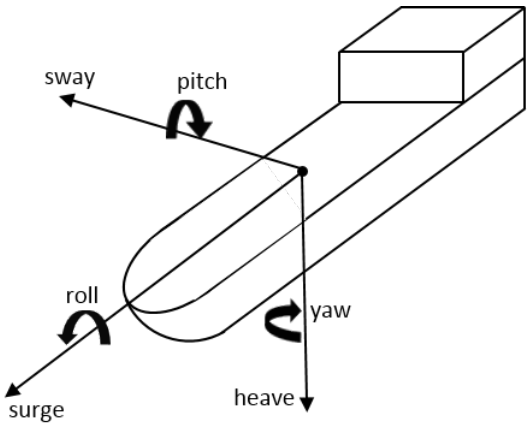


Figure 3.1: General motion of a vessel.

in the Earth-fixed frame is given by the generalized position vector

$$\boldsymbol{\eta} = \begin{bmatrix} n \\ e \\ d \\ \phi \\ \theta \\ \psi \end{bmatrix}. \tag{3.1}$$

The north-east-down positions n, e, and d and the rotation angles roll ϕ, pitch θ, and heading ψ are relative to the local Earth-fixed frame. The horizontal position in the Earth-fixed frame can be measured by GPS and gyrocompass (Perez, 2005).

Similarly, the body-fixed velocities in six DOF of a vessel is given by the generalized velocity vector

$$\boldsymbol{\nu} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}. \tag{3.2}$$

The linear velocities in surge u, sway v, and heave w and the angular velocities roll rate p, pitch rate q, and yaw rate r are expressed in the body-fixed frame (Perez, 2005). The linear body-fixed velocities are related to the local Earth-fixed frame by a linear-velocity transformation matrix

$$\boldsymbol{R}(\phi,\theta,\psi) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\psi c\phi s\theta \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}, \tag{3.3}$$

where $s \equiv \sin(\cdot)$ and $c \equiv \cos(\cdot)$ (Perez, 2005). Similarly, the angular body-fixed velocities are related to the Earth-fixed frame by an angular-velocity transformation matrix

$$\boldsymbol{T}(\phi,\theta) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}, \tag{3.4}$$

where $t \equiv \tan(\cdot)$ and $c\theta \neq 0$ (Perez, 2005). Both rotation matrices can be combined into one kinematic transformation matrix

$$\boldsymbol{J}(\phi,\theta,\psi) = \begin{bmatrix} \boldsymbol{R}(\phi,\theta,\psi) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{T}(\phi,\theta) \end{bmatrix}. \tag{3.5}$$

3.2 Kinetics and Hydrodynamic Forces

The study of ship dynamics can be divided into two different theories, depending on the field of study. The distinct difference between them is the way the effects of disturbances on the vessel are handled. The maneuvering theory covers the study of the motion of vessels in calm water at positive speed. Therefore, the excitation by waves is excluded in the maneuvering theory. This applies to e.g.

harbors and sheltered waters. However, if the motion of ships affected by waves is studied, seakeeping theory is applied. Here, zero or constant speed and a steady heading are assumed (Fossen, 2011).

Depending on the application, the model chosen to study the motion of the vessel or design control systems can origin from the maneuvering or the seakeeping theory. However, common practice is to combine the seakeeping model and the maneuvering model into a superposition model (Perez, 2005). The model uses superposition of either the force or motion. The LF motion, induced by the control action, is modeled by the maneuvering model and the motion induced by waves is modeled by the seakeeping model. In motion superposition, it is assumed that the WF disturbance can be treated as colored and additive measurement noise. This model is widely used for control system design. Contrary, in force superposition the WF disturbance is treated as process noise. Force superposition models are typically part of ship motion simulators (Perez, 2005).

The force superposition model of the vessel is represented by

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\phi, \theta, \psi)\boldsymbol{\nu}, \tag{3.6}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau}_{\text{control}} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{waves}}. \tag{3.7}$$

The inputs $\boldsymbol{\tau}_{\text{control}}$, $\boldsymbol{\tau}_{\text{wind}}$ and $\boldsymbol{\tau}_{\text{waves}}$ are the forces from the actuators and the wind and wave disturbances in the body-fixed frame. The velocity $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$ is the relative velocity, where $\boldsymbol{\nu}_c$ is the velocity of the ocean current (Fossen, 2011). In (3.7), $\boldsymbol{M}$ is the combined rigid-body and hydrodynamic added mass matrix. The added mass comes from the change in pressure of the water on the hull when accelerating and decelerating. These pressure induced forces are proportional to the acceleration and velocity relative to the water (Fossen and Perez, 2009). The matrix $\boldsymbol{C}(\boldsymbol{v}_r)$ is the Coriolis-centripetal matrix for the rigid-body and added mass (Perez, 2005). These terms appear when expressing the equations of motion in the body-fixed frame. Coriolis-centripetal terms can be ignored if the velocities are small, such as in positioning control (Fossen and Perez, 2009). The damping forces in (3.7) are represented by the nonlinear term $\boldsymbol{D}(\boldsymbol{\nu}_r)$. These forces reflect the transfer of energy from the vessel to the fluid. The term is a function of the speed and direction of the current relative to the vessel (Fossen, 2011).

3.3 Environmental Disturbances

A vessel is constantly affected by changing environmental disturbances. These environmental forces are wind, waves and ocean currents. Conceptually, they can be separated into low-frequency (LF) and oscillatory wave-frequency (WF) forces. LF forces, including wind and currents, cause drift, while WF forces make the vessel oscillate around the LF motion. When designing control systems, disturbances have to be treated differently depending on the application. In most DP applications, only the LF forces of the disturbances are accounted for, while in roll damping the WF motion is the disturbance to be mitigated (Fossen, 2011).

Wind and current disturbances can be modeled in a similar way. Both depend on the speed and direction of the disturbance relative to the vessel. Furthermore,

the projected area of the disturbance on the vessel are important factors. These areas are the submerged area of the hull for the current and the projected wind area for the wind disturbance. Moreover, they change at different relative angles of attack, are ship specific and can be difficult to estimate. Therefore, the LF disturbances wind and current are often modeled as constant or slowly varying (Fossen and Perez, 2009).

The disturbance induced by waves is divided into first order, linear wave disturbances, and second order, nonlinear wave disturbances, as

$$\boldsymbol{\tau}_{\text{waves}} = \boldsymbol{\tau}_{\text{wave}}^{\text{lin}} + \boldsymbol{\tau}_{\text{wave}}^{\text{nlin}}. \tag{3.8}$$

The linear wave disturbance is responsible for the WF disturbance. It can be considered to be stationary for periods from 20 minutes to 3 hours (Fossen and Perez, 2009). The dominating frequency of ocean waves is in the range of (Fossen, 2011)

$$0.3 \text{ rad/s} < \omega_0 < 1.3 \text{ rad/s}. \tag{3.9}$$

The nonlinear wave disturbance depends quadratically on the wave elevation. Hence, it affects the vessel with frequencies both lower and higher than the frequencies of the waves. The high frequencies, which have frequency content at the sum of the wave frequencies, are too high to be considered in vessel motion control. Therefore, the nonlinear wave disturbance, similar to wind and current, is commonly modeled as constant or slowly varying forces affecting the vessel (Fossen and Perez, 2009).

The linear wave disturbance is typically modeled as a linear approximation of wave-induced motion spectrum. Usually it is sufficient to approximate a spectrum with the second-order filter with relative degree one given by

$$H(s) = \frac{2\lambda\omega_0\sigma s}{s^2 + 2\lambda\omega_0 s + \omega_0^2}, \tag{3.10}$$

where λ is a damping coefficient, σ is the wave intensity, and ω_0 is the dominating wave frequency. One has to consider that each DOF is affected by the linear wave disturbance in different ways. The input to the shaping filter in (3.10) is Gaussian white noise (Fossen, 2011).

3.4 Wave Filtering

As discussed in Section 3.3, the disturbances can be divided into LF and WF disturbances. LF disturbances cause a drifting motion of the ship and has to be controlled to ensure good performance of the vessel. On the other hand, the oscillating linear WF disturbance with zero mean will cause the vessel to oscillate, but not affect the overall position-keeping performance. Figure 3.2 shows the total measured heading of a vessel affected by WF forces and the LF motion. Even though the vessel oscillates around the LF motion, the WF forces do not affect the overall heading. Hence, only controlling the LF motion is sufficient for DP and heading autopilot applications.

Usually the bandwidth of vessels are outside the frequency range of the WF disturbance (3.9). However, this does not apply to the bandwidth of the actuators

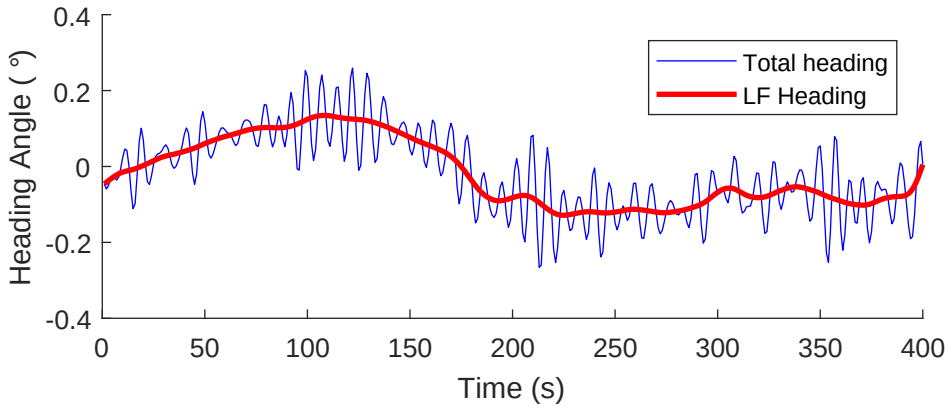


Figure 3.2: The LF motion and total heading of a vessel. The oscillating WF motion does not affect the overall heading.

and servos. Consequently, the actuators will start oscillating to compensate for the offset. By controlling the LF motion, excessive motion of the actuators, which are trying to correct for every single wave, is avoided. Therefore, fuel is saved and unnecessary wear of the actuators is prevented. To enable LF-motion-only control, wave filtering can be applied. Wave filtering refers to the removal of the WF motion from the position measurements and estimated velocities. Wave filtering, including WF-free state estimation, is an important feature in high-precision ship control (Fossen, 2011). Over time, several different methods have been developed to decrease the oscillating movement of the actuators or rudders.

An early approach, used in adaptive control, imposes a deadband of the actuator or rudder. Implementing a deadband in the control system allows for some deviation from the setpoint without the controller reacting (Fossen and Perez, 2009). This approach was, for instance, used in van Amerongen (1985) in the implementation of an adaptive controller.

Low-pass and notch filters have been applied to eliminate the oscillating motion (Schei, 1995). Low-pass filters are sufficient when the bandwidth of the controller is much lower than the frequency of the disturbance. That is the case for large ship, such as oil tankers. If the bandwidth of the controller is close to, or overlaps with, the bandwidth of the disturbance, a notch filter has to be used in cascade with the low-pass filter (Fossen, 2011). A disadvantage with using conventional filters is the introduction of significant phase-lag into the feedback. This can in turn affect the performance of the control system.

More advanced techniques, such as the KF, have been used in DP systems since around 1975 (Schei, 1995). These techniques are model based state estimators and assume motion superposition. A good model of the system and the disturbance is crucial for good wave filtering performance. Balchen et al. (1980) proposed a DP system, based on the KF, to get optimal estimation of the vessel motion and environmental forces. More information about the KF design for DP can be found in Fossen and Perez (2009). Schei (1995) compared a stationary KF to a generalized notch filter and comes to the conclusion that a conventional notch filter can achieve the same performance as the stationary KF. The study also showed that the dynamics in the feedback loop for an observer based approach

and a conventional notch filtering approach are approximately the same.

An alternative method for wave filtering of nonlinear systems are nonlinear passive observers. This was proposed for DP systems by Fossen and Strand (1999). Nonlinear passive observers have the advantage, compared to a KF, of guaranteeing global convergence of all estimation errors to zero without having to estimate any covariance matrices. This is achieved by computing observer gains, which are peak wave frequency depended, through passivity theory (Fossen, 2011) .

Lindegaard and Fossen (2001) proposed a wave filtering state estimator by taking advantage of the developing sensor technology of integrated navigation systems. These systems integrate GPS and inertial measurement units to measure velocity and linear acceleration with great accuracy. The authors created a combined wave filtering and estimation system using all available data.

Except for the deadband approach, the dependency of correctly estimated wave features, like the wave spectrum or the wave peak frequency, is a common factor in all wave filtering approaches. Hence, frequency estimators have to be implemented. Belleter et al. (2013) proposed a nonlinear wave encounter frequency estimator. The estimator considered is designed to estimate the frequency, amplitude and phase of a unknown sinusoid using measurements of the roll angle of the ship. Other methods of obtaining the wave frequency is by power spectral analysis (Enshaei and Birmingham, 2012) or by estimating a directional wave spectra (Nielsen, 2006).

4 Learning a LF-Motion Predictor

In this chapter, the predictor algorithm is introduced. The approach creates a framework around a recurrent neural network that enables an iterative process of NN training and position prediction.

4.1 Neural Network for LF-Motion Prediction

As discussed in Chapter 2, a NN, which is trained by supervised learning, needs input data and target data for training. After proper training, the NN should be able to map new inputs to new predicted outputs. Therefore, the goal of this algorithm is to predict new LF positions with the NN that has previously been trained with input and target data. The available data is the sensor data from GPS and gyrocompass and logged data of forces applied by the actuators. This data alone should contain information about the dynamics of the ship and the behavior of the disturbance. The available data can be defined as the data set

$$\mathcal{Z}^M = \{u(\tau), y(\tau)\}_{\tau=t}^{t-M+1}, \tag{4.1}$$

where M is the total amount of collected samples, the control signal $u(\tau) \in \mathbb{R}^k$ and the measured positions $y(\tau) \in \mathbb{R}^l$. The number of control input sequences and measured data sequences are denoted k and l, respectively. The current available data $\mathcal{Z}^M$ is the data that can be used for prediction.

The problem lies in specifying the target data. In the case of wave filtering, the desired output, the LF position, is unmeasurable or too expensive to measure. It can be seen as the mean of the oscillating position. The fact that the desired output of the NN does not exist physically creates a problem the algorithm has to deal with, i.e. how to create the target data.

A NN is known as a good function approximator, as described in Chapter 2. To get a well approximating and generalizing NN, large amount of data is usually required. However, the question remains - what happens if the system changes due to changing disturbances, as is the case in dynamic positioning? If the system changes, the NN trained on target data created before the change will not be able to map the input to the correct desired output anymore. Figure 4.1 shows the prediction error over time while the dominating wave frequency changes and the NN does not adapt. The changing nature of the disturbance leads to an extension of the problem that the algorithm has to encounter; how to create target data representing the newest disturbance and how to update the NN according to the change.

The restrictions of the target data and the input data can be specified even more. The target data does not only have to represent the newest disturbance, it is also not allowed to represent too old disturbances. If the network was trained on all disturbances the vessel had been affected by during a long period, the NN would learn to filter out the whole bandwidth of encountered disturbance frequencies. This is not desired, as mentioned in Section 3.4. The bandwidth of the

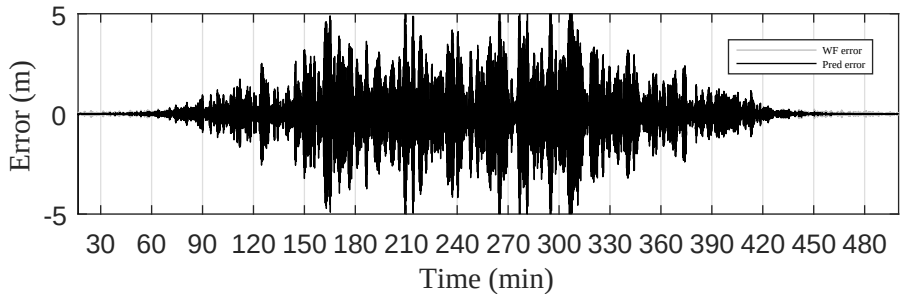


Figure 4.1: Prediction error for a NN predicting the LF motion. The NN is trained on data with a wave disturbance with a dominating frequency at 1.25 rad/s. The frequency of the disturbance in the data used for prediction changes from 1.25 rad/s to 0.3 rad/s and back to 1.25 rad/s. The NN is not able to predict the LF motion when unknown disturbances occur.

controller can overlap with the bandwidth of the WF disturbance. Nevertheless, it is not beneficial to compress the target data too much since the dynamics of the vessel still have to be present in the target data.

The tasks of the LF-motion predictor can be summarized as:

1. Create target data which represents the newest disturbance but also the dynamics of the vessel.
2. Update the NN if changes have occurred.
3. Use the NN to predict the LF motion of the vessel.

4.2 Creating Target Data

The target data has to be created out of the known data set $\mathcal{Z}^M$. If M is a large positive number, $\mathcal{Z}^M$ will contain a lot of information about the system. This will favor the learning of the system dynamics. This also entails that old disturbance dynamics are represented, which do not favor accurate LF-position prediction. The data collected for training should contain enough information about the system to make an approximation of the system possible, but also contain as current disturbance information as possible. Thus, a trade-off between system and disturbance has to be made. The input data for training will be defined as the training data set

$$\mathcal{Z}^N_{TR,t} = \{u(\tau), y(\tau), y_T(\tau)\}_{\tau=t}^{t-N+1}, \tag{4.2}$$

where $N \leq M$ and N defines the number of samples used for training and $y_T(\tau) \in \mathbb{R}^l$ is the target data. The data set $\mathcal{Z}^N_{TR,t}$ is a moving window, which always represent the last N values.

The following step is to create the corresponding target data

$$y_T(\tau),\ \tau = t, \ldots, t-N+1.$$

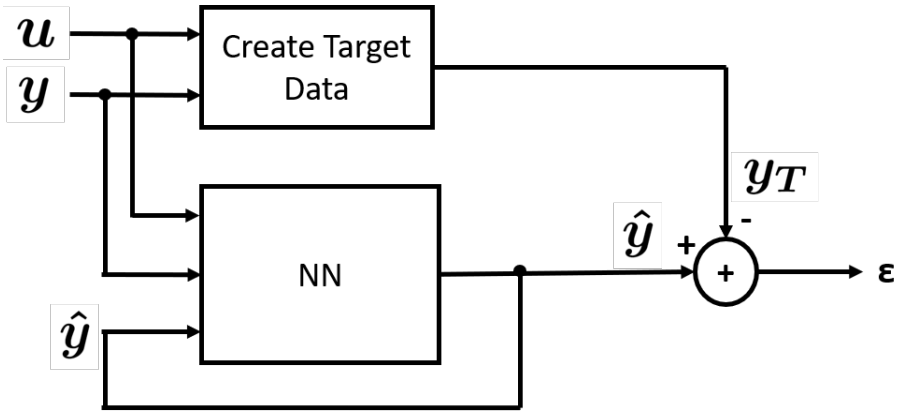


Figure 4.2: Schematic of the training of a RNN. The error ε is minimized during training. The data needed for training is defined by $\mathcal{Z}^N_{TR,t}$, where the target data $y_T(t)$ is created from the available data $u(t)$ and $y(t)$.

Figure 4.2 illustrates the rough schematic of the supervised training of a RNN. The figure shows that the target data is created out of the known data.

Several signal processing tools are applied to create the target data out of the measured position data. The aim is to filter the measured position in order to generate a filtered data sequence that can be used as target data. This filtering requires that the frequency of the oscillating disturbance is approximately known. Hence, the dominating peak frequency is estimated before filtering. The part of the LF-motion predictor that creates training data can be divided into three different areas: Data definition, frequency estimation, and filtering. The data definition was presented in this section. In Figure 4.3, the operation of the predictor is shown.

Figure 4.4 shows the target data needed at different times. The constant Δ defines the time between two NN training sessions. The figure illustrates that for every new network training, new target data is required to represent the newest disturbance dynamics.

4.2.1 Frequency Estimation

To create new target data, the first step is to estimate the peak frequency of the current wave disturbance. Estimation is performed on

$$y_{\text{est}}(t) = \{y(\tau),\ \tau = t, \ldots, t-\gamma\}, \tag{4.3}$$

which consists of past measured position data and is marked with red color in Figure 4.5. Since past data sequences are used to create training data, offline methods for frequency estimation can be applied.

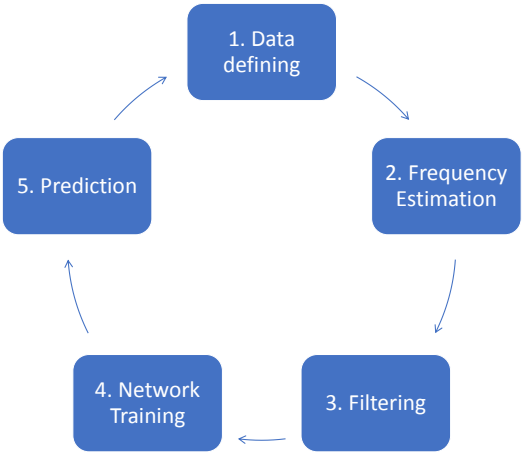


Figure 4.3: Schematic of the proposed algorithm. After defining the data available for training, the target data is created by estimating the dominating frequency and filtering the data. The target data is used to update the weights of the NN and the next LF positions are predicted.

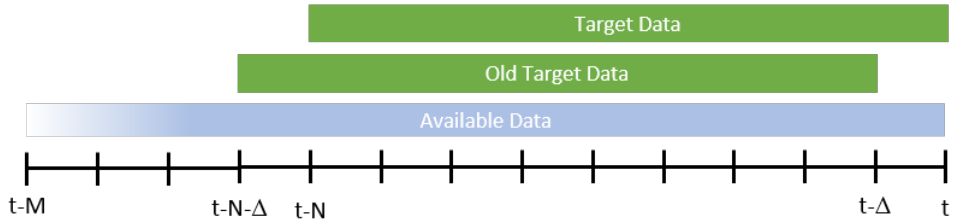


Figure 4.4: The target data is created from the available data and has to be updated for every new NN training.

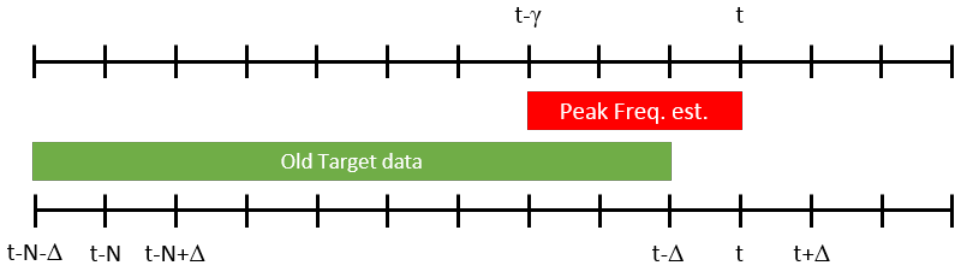


Figure 4.5: The wave disturbance peak frequency has to be estimated from recent data.

A wave disturbance has its dominating frequency in the range of 0.05 Hz to 0.2 Hz (Fossen, 2011). Large vessels have their bandwidth much lower than 0.05 Hz and the disturbance can be eliminated by using a low-pass filter with a cutoff frequency between the bandwidth of the vessel and the bandwidth of the wave disturbance. For smaller vessels, the controller bandwidth can be close to or within the bandwidth of the wave frequency. In those cases, Fossen (2011) presents the use of a low-pass filter in cascade with a notch filter, where the notch frequency of the filter is chosen to be the peak frequency of the wave disturbance if the speed of the ship is zero. In case of a moving ship, the notch frequency should be the encounter frequency.

For adaptive applications, the encounter frequency can, for example, be estimated by nonlinear signal-based wave encounter frequency estimators (Belleter et al., 2015) or the Sliding Fast Fourier Transform spectral analysis (Enshaei and Birmingham, 2012). In this study, the sea state, and therefore the encounter peak frequency, is assumed to be stationary for 20 minutes, see Section 3.3. In other words, the peak frequency does not have to be updated every second and the frequency can be estimated on back-dated data. Therefore, spectral analysis methods can be used.

The FFT allows deeper studies of the training data by transforming a signal from the time domain to the frequency domain. One can see in Figure 4.6 that the dominating frequency component is the low-frequency motion of the system. A smaller peak can be found at the wave disturbance frequency. To enable estimation of the peak frequency of the disturbance ω_0, a low-order high-pass filter is applied to the estimation data $y_{\text{est}}(t)$. The dominating frequency component of the remaining data should be the peak frequency of the wave disturbance. Here, a second-order Butterworth high-pass filter with a cutoff frequency ω_{HP} inside the bandwidth of the wave disturbance should be sufficient to eliminate the residues of the low frequency motion of the model and still enable the disturbance to be estimated. The parameter γ, which defines the number of past samples used to create the frequency spectrum, has to be chosen so that the desired resolution of the spectrum is achieved.

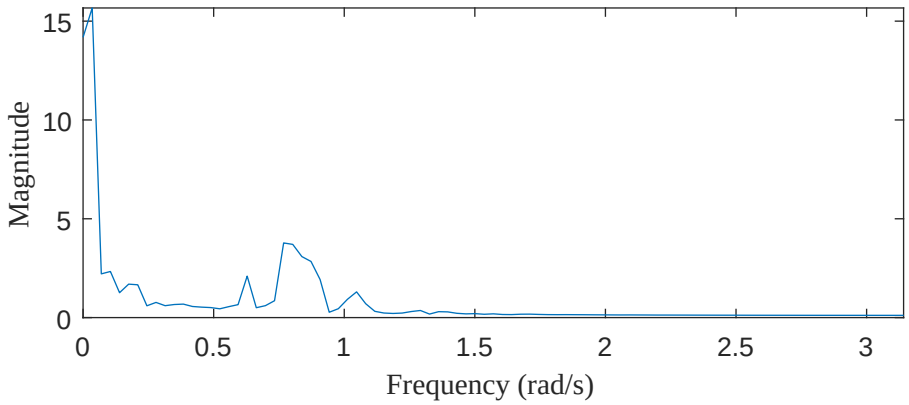


Figure 4.6: Frequency content of a sequence with 180 samples of yaw data from a ship model. The dominating wave frequency is $\omega_0 = 0.9$. The dominating frequency can be identified by high-pass filtering the sequence.

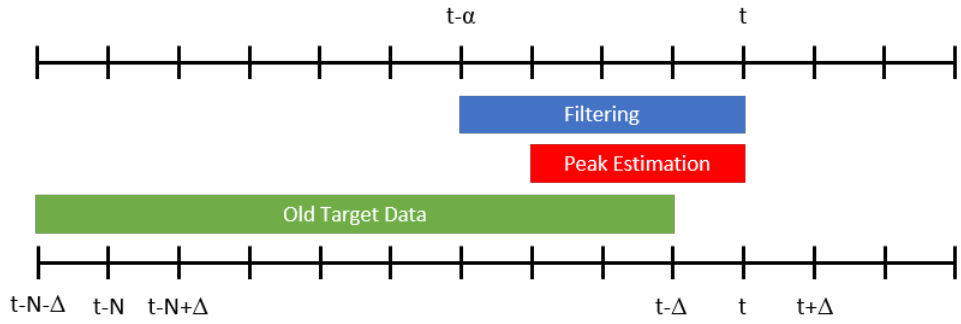


Figure 4.7: Filtering of the measured position after estimating the dominating wave frequency.

To estimate the frequency after high-pass filtering, the spectral power density is calculated using Welch's method. To estimate the power spectral density, the sequence is divided into overlapping sections. These sections are multiplied with a window function and modified periodograms are created from them using the fast Fourier transform. The periodograms are then averaged, which leads to reduced variance of the power spectral density estimation (Welch, 1967). For the estimation the implemented function `pwelch()` from Matlabs Signal Processing Toolbox was used and a Hamming window was applied.

4.2.2 Filtering

The filtering, in combination with the frequency estimation, is an important part of the algorithm. Together, they create the target data and thereby set an upper limit on how well the network can be trained. The goal of the frequency estimation is to find the encounter frequency of the wave disturbance. This frequency is needed to design filters to eliminate the disturbance. Figure 4.7 shows how the measured positions

$$y(\tau),\ \tau = t, \ldots, t-\alpha$$

is chosen for filtering.

Filtering $y(t)$ removes most of the oscillating disturbance and keeps as much necessary information of the LF motion as possible. The filtering will create the filtered data

$$y_{\mathrm{f}}(t) = H_{\mathrm{f}}(q)y(\tau), \tau = t, \ldots, t-\alpha$$

where $H_{\mathrm{f}}(q)$ denotes the filter. One wave filtering approach applies a notch and low-pass filter in series (Fossen, 2011). The estimated peak frequency is needed to design the notch filter, where the estimated frequency defines the notch frequency ω_{notch}. The task of the low-pass filter, with a cutoff frequency at ω_{LP}, is to eliminate high-frequency noise. Consequently it does not affect the target creation as much as the notch filter. The notch filter has to be well tuned to achieve acceptable target data. Tuning-parameters are the depth and the width of the notch. A too wide notch potentially eliminates information about system dynamics. A too narrow notch is not able to eliminate all of the disturbance and is less robust for model errors. Careful parameter tuning is, thus, essential for a

well performing algorithm.

The filter parameters chosen for the notch and low-pass filters are listed in Table A.3 and the corresponding frequency response at $\omega_0 = 0.3$ rad/s can be seen in Figure 4.8.

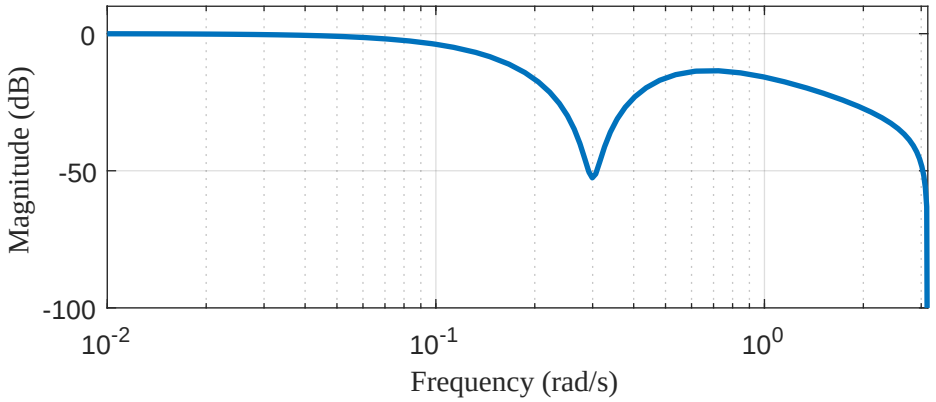


Figure 4.8: Bode diagram of a notch and low-pass filter in series. The notch frequency $\omega_{\text{notch}} = 0.3$ rad/s and the cutoff frequency $\omega_{\text{LP}} = \omega_{\text{notch}} + 0.2$ rad/s.

Note that it is the target data used for network training that is created through the filtering, whilst the cascade filtering, for instance, described in Fossen (2011), creates the LF-position estimate directly. Due to the advantage of offline computation with back-dated data, the designed notch and low-pass cascade filter is applied using a noncausal filtering method. This will eliminate phase lag, which would be introduced during online filtering with the cascade filters. A behavior of the NN, comparable to a noncausal filter is desired. One noncausal filtering method is forward-backward filtering. This method eliminates the time delay by re-filtering the filtered data backwards. Even though the initial values can be optimized and thereby transients minimized, some deviation from the actual LF position can be noticed at the beginning and the end of the sequence. An example of transients is seen in Figure 4.9, where the filtered motion is compared to the actual LF motion.

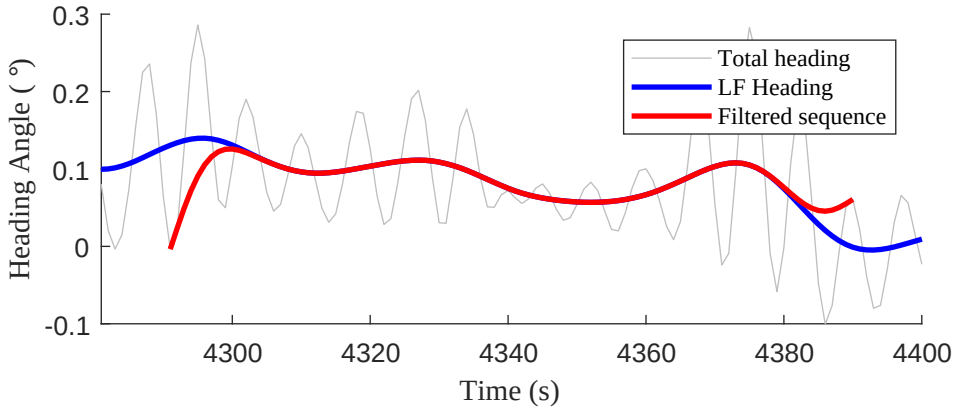


Figure 4.9: Example of transients appearing during forward-backward filtering.

To eliminate the transients, overlapping of data is applied. The overlapping is defined by the transient length ϵ, which is the maximum numbers of samples that have been affected by transients. This parameter can be used to define the length of the filtered data α, see Figure 4.7. The length of the filtered data should be large enough to eliminate the transients at the end of the old target data $y_T(t-\Delta)$ and the beginning of the current filtered data $y_\mathrm{f}(t)$. Therefore can α be defined as

$$\alpha \geq \Delta + 2\epsilon. \tag{4.4}$$

The part of the filtered data $y_\mathrm{f}(t)$ that is appended to the old target data $y_T(t-\Delta)$ can be denoted as the appending length β, see Figure 4.10. The appending, as seen in Figure 4.10, eliminates the transients at the beginning of $y_\mathrm{f}(t)$ and the end of $y_T(t-\Delta)$ if

$$\Delta + \epsilon \leq \beta < \alpha - \epsilon. \tag{4.5}$$

After appending, the first Δ elements of the old target data $y_T(t-\Delta)$ are discarded. Then the new target data $y_T(t)$ is prepared for training, see Figure 4.10.

The overlapping procedure presented does not eliminate possible transients at the end of the target data $y_T(t)$. To avoid training on deviating data, the error of the last ϵ samples of the training data are neglected. This avoids training on erroneous target data.

4.3 Neural Network Training

When training a NN under supervision, an input sequence and a target sequence are presented to the NN as input and output, respectively. The reason for training is to adjust the weights of the network in such a manner that the NN is able to map the input sequence to the output sequence. After the target data $y_T(t)$ is created, the network can be trained using the training set $\mathcal{Z}^N_{TR,t}$. The procedure of appending new data to already existing target data will lead to target data with filter sequences of length Δ. Thus, all previous filtered sequences $y_\mathrm{f}(t-\Delta), y_\mathrm{f}(t-2\Delta), \ldots$ are represented with Δ samples up to $y_\mathrm{f}(t-N+\Delta)$, see Figure 4.11.

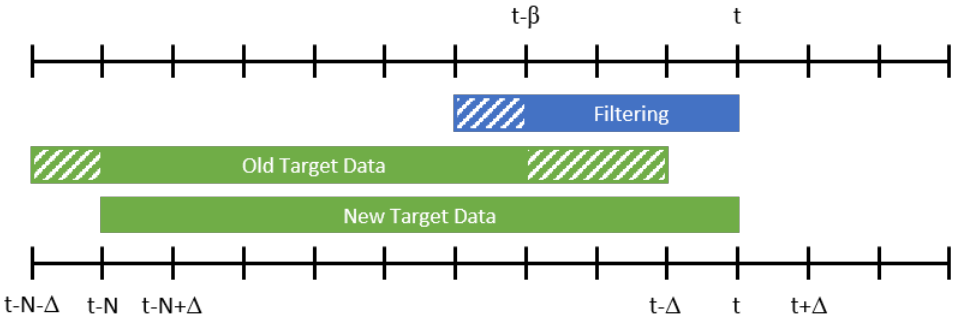


Figure 4.10: Appending filtered data to old target data to create new target data.

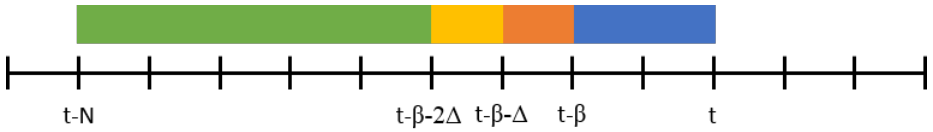


Figure 4.11: The target data consists of Δ samples of past filtered data.

As long as the relation between the input and output sequence does not change and the values of the input sequences are close to the values used while training, a well trained NN will be able to successfully map the input to the desired output. If the relation between the input and the desired output changes through, for example, drift or changing wave disturbances, the NN is not able to map the input to the correct output anymore, see Figure 4.1. This entails that the weights of the NN have to be adjusted. Therefore, the repetitive NN training is the core of the algorithm described in this chapter. The training has to ensure correct execution of the networks task, to process the data available at time t and predict the next LF position $y_{\mathrm{LF}}(t+1) \in \mathbb{R}^l$.

Training the NN will adjust the weights of the network and thereby prepare it to deal with the current characteristics of the disturbances. The challenge when training the NN with the prepared data windows is the risk of overfitting on the short data sequence. To avoid overfitting, the number of epochs, which describes the number of backpropagations and weight updates per training, can be limited. Also, increasing the value of parameter N, which defines the length of the training data set, will lead to a lower risk of overfitting. Furthermore, regularization can be applied to improve generalization. Early stopping, which is a simple solution is not suitable for this approach because of the changing behavior of the disturbance and that all data is needed for training.

For this particular study, a NN with a RNN structure is chosen. More specific a NNARMAX structure, which also uses past measured positions as input and therefore can learn the nature of the disturbance.

4.4 Prediction

The last step of the algorithm is to use the trained NN to predict the LF position. Depending on the configuration of the NN, different length of data is required. Similar to a discrete transfer function, the output of the NN is dependent on a limited amount of previous inputs and, in case of the RNN, outputs. The data needed to predict $\hat{y}_{\mathrm{LF}}(t+1)$ with a NN can be defined as the control signal

$$u_{\mathrm{p}}(t) = \{u(\tau), \tau = t, \ldots, t - n_{\mathrm{u}} + 1\},$$

the measured position

$$y_{\mathrm{p}}(t) = \{y(\tau),\ \tau = t, \ldots, t - n_{\mathrm{y}} + 1\},$$

and the feedback

$$\hat{y}_{LF,\mathrm{p}}(t) = \{y(\tau), \tau = 1, \ldots, t - n_{\mathrm{fb}} + 1\},$$

where n_{u}, n_{y}, and n_{fb} are the numbers of delayed control inputs, measured positions and feedback values for the NN. The predicted LF position $\hat{y}_{\mathrm{LF}}(t+1)$ is then used by the controller to calculate the next control input. In an ideal case, the LF prediction should not contain any of the wave disturbance frequencies and therefore only represent the LF movement of the vessel.

When using the NNARMAX model, the LF-position prediction depends on the previous measured position. Thus, only one-step prediction is possible. However, the amount of one-step predictions is not limited. To clarify, it is possible to make one-step predictions for an unlimited amount of time before the network is trained again. A too long period without training could affect the prediction, as seen in Figure 4.1.

To measure the performance of the predictor, the difference between the filtered target data and the prediction is analyzed. The difference should be small or at least lie within the difference between target data and measured position.

4.5 Algorithm

To simplify re-creation of the proposed approach, a pseudocode of the procedure discussed in this chapter is presented in Algorithm 4.1. The algorithm uses the notation introduced in this chapter.

Algorithm 4.1 Method for adaptive training of LF-motion predictor

Define n_{u},n_{y},n_{fb} and create the RNN
Define γ,Δ,N,ϵ
Require: $\alpha \geq \Delta + 2\epsilon$
Require: $\Delta + \epsilon \leq \beta < \alpha - \epsilon$
Define:
$\mathcal{Z}^{N}_{TR,t} = \{u(\tau), y(\tau), y_T(\tau)\}_{\tau=t}^{t-N+1}$
$y_{\mathrm{est}}(t) = y(\tau),\ \tau = t, \ldots, t-\gamma$
$u_{\mathrm{p}}(t) = \{u(\tau), \tau = t, \ldots, t-n_{\mathrm{u}}+1\}$
$y_{\mathrm{p}}(t) = \{y(\tau),\ \tau = t, \ldots, t-n_{\mathrm{y}}+1\}$
$\hat{y}_{LF,\mathrm{p}}(t) = \{y(\tau), \tau = 1, \ldots, t-n_{\mathrm{fb}}+1\}$
Initialize net
Initialize $\hat{y}_{LF,\mathrm{p}}(t) \leftarrow y(\tau), \tau = t, \ldots, t-n_{\mathrm{fb}}+1$
$t \leftarrow 1$
repeat
    $t \leftarrow t+1$
    function PREDICTOR(net,$u_{\mathrm{p}}(t)$,$y_{\mathrm{p}}(t)$,$\hat{y}_{LF,\mathrm{p}}(t)$)
        Use the NN to predict the next LF-position
        return $\hat{y}_{\mathrm{LF}}(t+1)$
    end function
    if $(t \bmod \Delta) = 0$ **then**
        function FREQUENCY ESTIMATION($y_{\mathrm{est}}(t)$)
            Apply a high-pass filter with a cutoff frequency at ω_{HP}
            Estimate the power spectral density and find the dominating frequency
            return ω_0
        end function
        function FILTERING($y(t)$,ω_0)
            Design notch filter with a notch frequency at ω_0
            Design a low-pass filter with a cutoff frequency at ω_{LP}
            Apply the filters in series on $y(\tau),\ \tau = t, \ldots, t-\alpha$
            return $y_{\mathrm{f}}(t)$
        end function
        function APPENDING($y_{\mathrm{f}}(t)$,$y_T(t-\Delta)$)
            Delete the newest $\beta - \Delta$ values of $y_T(t-\Delta)$.
            Delete the oldest Δ values of $y_T(t-\Delta)$.
            Append the newest β values of $y_{\mathrm{f}}(t)$.
            return $y_T(t)$
        end function
        function NETWORK TRAINING(net,$\mathcal{Z}^{N}_{TR,t}$)
            Train the RNN using $\mathcal{Z}^{N}_{TR,t}$
            Neglect the error of the last ϵ samples
            return net
        end function
    end if
until end of usage

5 Simulations and Results

This chapter includes a presentation of the simplified models used for simulation, a description of the conducted experiments as well as a discussion of the results.

The developed algorithm is evaluated on a variation of simulations. During one simulation the neural network is created, the motion of the vessel is simulated, and the prediction algorithm, as discussed in Chapter 4, is applied. The complexity of the simulations was increased over the course of this study. The RNNs used during the simulations are created, trained and applied using Matlabs Neural Network Toolbox.

The parameters used for the algorithm can be reviewed in Appendix A. Most parameters were chosen by a trial and error approach and do not represent optimal values. Rather than finding one optimal solution, the aim of the presented study is to explore the possibilities of the proposed algorithm.

5.1 Simplified Simulation Models

To assess performance of the LF predictor presented in Chapter 4, it is applied to different nonlinear systems with increasing complexity. The complexity is increased from a simple SISO system using one DOF, over a MIMO system using three DOFs to a full ship model. The simplified models are supposed to describe the underlying dynamics of the vessel and the wave disturbance. The wave disturbance is included into the models following the force superposition model. Most focus was put on the first-order wave disturbance in the simulations since wave filtering is considered.

5.1.1 Wave-Frequency Disturbance

This section presents the model of the WF disturbance used for simulation. Different system models were used while the same wave disturbance model was used for all systems. To simulate the wave disturbance, a linear approximation of a wave spectrum is realized. One example is the JONSWAP spectrum, which is the result of an extensive measurement program in the North Sea and it can be used to describe non-fully developed sea. The JONSWAP spectrum can be sufficiently approximated by a second-order linear transfer function, see Section 3.3. For motion superposition, a sufficient approximation is

$$\boldsymbol{\eta} = \boldsymbol{\eta}_{\mathrm{LF}} + \boldsymbol{K}_{\mathrm{w}}\boldsymbol{G}(s)\boldsymbol{e}(s), \tag{5.1}$$

where $\boldsymbol{K}_{\mathrm{w}}$ is a fixed gain, $\boldsymbol{G}(s)$ is a second-order approximation of the wave spectrum and $\boldsymbol{e}(s)$ is white noise (Fossen, 2011). In force superposition, a motion-to-force transformation has to be applied. To simplify, a shaping filter of sixth

order was implemented to recreate the behavior described in (5.1) as

$$H_{\mathrm{HO}}(s) = \frac{K_\omega s^2}{(s^2 + 2\lambda\omega_0 s + \omega_0^2)^3}, \tag{5.2}$$

where the the gain is

$$K_\omega = 2\omega_0^3 \lambda\sigma, \tag{5.3}$$

ω_0 is the dominating wave frequency, λ is a damping coefficient and σ describes the wave intensity. This shaping filter gives the force affecting the vessel as

$$\boldsymbol{\tau}_{\mathrm{waves}}^{\mathrm{lin}} = H_{\mathrm{HO}}(s)e(s). \tag{5.4}$$

During the simulation, the dominating wave frequency is varied over the frequency range (3.9), while the damping coefficient and the wave intensity are held constant.

Using the cube of the peak frequency in the gain K_ω the gain for higher frequencies is increased. This enables using the wave disturbance as input disturbance and applying the force superposition model.

5.1.2 One DOF Model

The model of the system representing the vessel is kept as simple as possible. The most straightforward way is a model with one DOF. The system is modeled as

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a x_2 |x_2| + b(\boldsymbol{\tau}_{\mathrm{c}} + \boldsymbol{\tau}_{\mathrm{wave}}^{\mathrm{lin}}) \\ y = x_1 \end{cases}, \tag{5.5}$$

where $\boldsymbol{\tau}_{\mathrm{wave}}^{\mathrm{lin}}$ is the wave disturbance and $\boldsymbol{\tau}_{\mathrm{c}}$ is the control input. To get a suitable frequency response with a crossover frequency between 0.01 rad/s and 0.05 rad/s, the model parameters where chosen as

$$a = -0.017 \tag{5.6}$$

and

$$b = 0.00068. \tag{5.7}$$

A step response of the system can be seen in Figure 5.1. For comparison, the response of a linear system with a cross over frequency at around 0.03 rad/s is also shown.

5.1.3 Three DOFs Model

A slightly more realistic model with three DOFs is given by

$$\begin{cases} \dot{\boldsymbol{x}}_1 = \boldsymbol{J}_{\mathrm{3DOF}}(\psi)\boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 = \boldsymbol{A}\boldsymbol{x}_2|\boldsymbol{x}_2| + \boldsymbol{B}(\boldsymbol{\tau}_{\mathrm{c}} + \boldsymbol{G}(\psi)\tau_{\mathrm{wave}}^{\mathrm{lin}}) \\ \boldsymbol{y} = \boldsymbol{x}_1 \end{cases}, \tag{5.8}$$

where the rotation matrix in (3.3) is stripped down to

$$\boldsymbol{J}_{3\text{DOF}}(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.9}$$

to represent only the three DOFs surge, sway, and yaw rate. The system parameters are defines as

$$\boldsymbol{A} = \begin{bmatrix} -0.017 & 0 & 0 \\ 0 & -0.017 & 0 \\ 0 & 0 & -0.00001 \end{bmatrix}, \tag{5.10}$$

and

$$\boldsymbol{B} = \begin{bmatrix} 0.00068 \\ 0.00068 \\ 0.00001 \end{bmatrix}. \tag{5.11}$$

Here, $\boldsymbol{x}_1 = [n, e, \psi]^T$ represents the position in the Earth-fixed frame and $\boldsymbol{x}_2 = [u, v, r]^T$ represents the velocities in the body-fixed frame. The nonlinear rotation matrix $\boldsymbol{J}_{3\text{DOF}}(\psi)$ transforms the position from the body-fixed frame to the Earth-fixed frame. The rotation matrix $\boldsymbol{G}(\psi)$ makes the effect of the WF force on surge, sway, and yaw depending on the relative heading angle. It is defined as

$$\boldsymbol{G}(\psi) = \begin{bmatrix} K_{\text{surge}} \cos(\psi) \\ K_{\text{sway}} \sin(\psi) \\ K_{\text{yaw}} \sin(\psi) \cos(\psi) \end{bmatrix}, \tag{5.12}$$

where K_{surge}, K_{sway} and K_{yaw} are gains for the disturbances in surge, sway, and yaw.

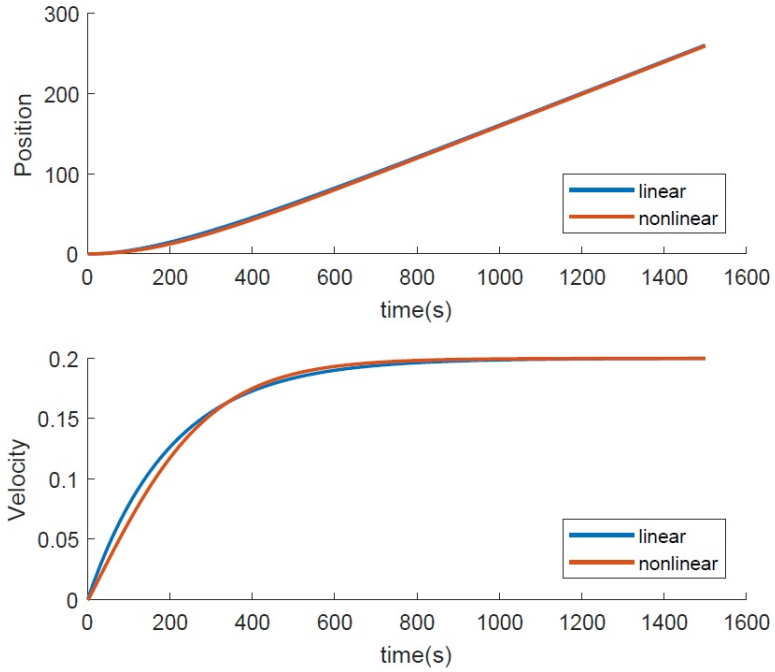


Figure 5.1: Step response of linear and nonlinear model according to (5.5).

5.2 One DOF system simulation

To evaluate if the proposed predictor is able to predict the LF motion at all, first simulations are performed using the one DOF model (5.5). The simulations have a length of 30 000 samples at a sample frequency of 1 Hz. The dominating frequency of the disturbance changes from 1.25 rad/s at time 0 s to 0.3 rad/s at 10 000 s. The disturbance frequency is held constant until 20 000 s before the frequency is decreased again to 1.25 rad/s at 30 000 s. Over a period of 20 minutes, the frequency changes with 0.114 rad/s. The illustrated graphs start at $N+1$, where N is the length of the data window. All predictions are executed ten times on ten different motion scenarios. The networks are initialized with the same small weights. The system is excited by bounded Gaussian white noise with a bandwidth from 0 rad/s to 0.2 rad/s.

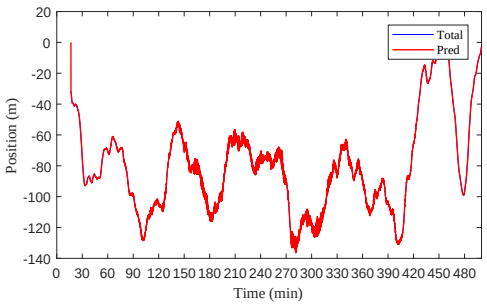
For every set of ten runs, the run with the worst prediction is removed. This prevents an incorrect interpretation of the result due to one bad simulation. A flawed simulation can occur due to the lack of convergence or the fact that it is not yet possible to guarantee that the predictor always learns the correct model with the implemented training method. Hence, the resulting prediction error of a bad converging or a wrongly learned model is large enough to overshadow the remaining results. This can turn out as, for example, a wildly oscillating prediction. In online use it would be possible to detect an incorrect predicting predictor and reset the system. However, this feature has not been implemented in this study.

To be able to compare the results, the root mean square of the error (RMSE) is calculated for every run. It is calculated for the interval $5\,000 \text{ s} \leq t \leq 30\,000$ s. After removing the worst run from every set, the mean RMSE (MRMSE) for the remaining runs is calculated and the standard deviation (STD) of the error over time is illustrated.

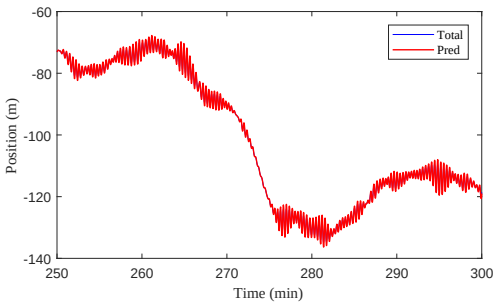
5.2.1 Reference Simulation

In order to evaluate the results of the adaptive LF-motion predictor, a reference is needed. The reference is created by predicting the LF motion without updating the weights of the RNN while the disturbance changes. The error between target and prediction data can later be used to compare LF predictions where the weights have been updated adaptively.

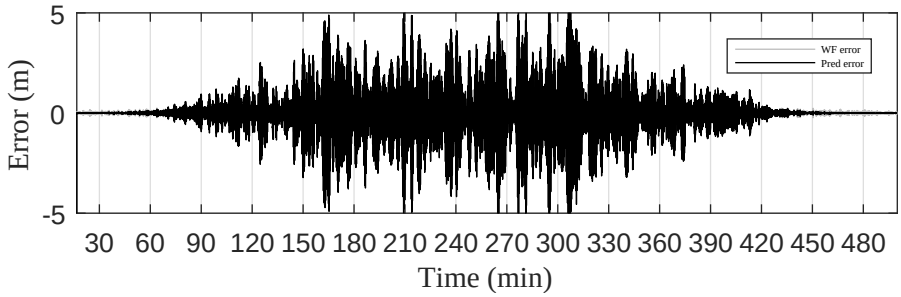
To achieve a reference, a RNN is created and the one DOF system is simulated. Before applying the LF-motion prediction algorithm, the NN is trained on a 30 000 sample data set with a constant wave disturbance frequency of 1.25 rad/s. This procedure of training the RNN offline will be referred to as pretraining. The pretrainined RNN is then used to predict the LF motion of another data set with changing wave peak frequency. The resulting prediction of the LF motion is shown in Figure 5.2. One can see that the LF-motion prediction overlaps almost perfectly with the oscillating motion of the vessel. In Figure 5.2c, the difference between the target data and the LF-motion prediction is shown. This figure displays how the error changes along with the change of the oscillating disturbance.



(a) Total motion and LF prediction.



(b) Enlarged image of the total motion and LF prediction.



(c) Prediction error compared to the oscillating disturbance. The MRSME of the prediction is MRSME = 1.2633

Figure 5.2: Reference Simulation: A static pretrained NN is not able to predict the LF motion when the disturbance changes.

The STD of the error of nine runs can be seen in Figure 5.3. This indicates that a NN trained on a static disturbance is not able to predict the LF motion if the nature of the disturbance, as in this case the dominating frequency and the amplitude, changes.

5.2.2 One DOF LF-motion Prediction in Open Loop

To show the advantage of the LF-motion predictor, to be able to learn a model without knowledge about the system, the pretraining procedure is discarded. Hence, the algorithm has to learn the system from random initialization, while predicting the LF motion. Other than using the proposed Algorithm 4.1, the simulation setup is the same as in Section 5.2.1.

In Figure 5.4, one LF-motion prediction is compared with the respective total motion. The LF prediction shows that it is possible to use the proposed algorithm to predict the LF motion with a reduced amplitude of the oscillating motion. Furthermore, the difference between the target data and the LF-motion prediction is illustrated in Figure 5.4c and compared to the difference between the target data and the total motion. Again, one can see that the magnitude of error from the LF motion is reduced. A comparison of the MRMSEs ($0.1468 < 1.2633$) also indicates a reduction of the oscillating motion.

These results indicate that the NN is able to learn a model to predict the LF motion. The STD of the error in Figure 5.5 show that the STD stays small

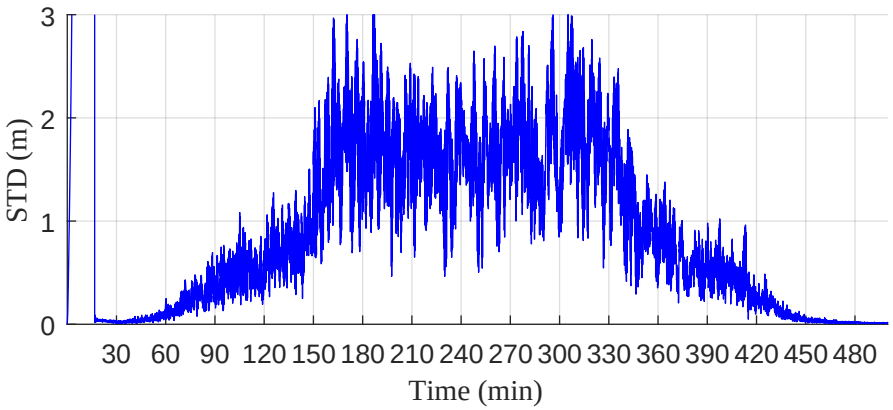


Figure 5.3: STD of the prediction error for nine one DOF simulations for a pretrained network without update. MRMSE = 1.2633

over the whole sequence. This indicates that the algorithm is able to suppress oscillating disturbance with high amplitude. However, when the disturbance is small the prediction can get imprecise. A deviation from the LF motion even though the oscillating disturbance is small indicates that the model could be improved further.

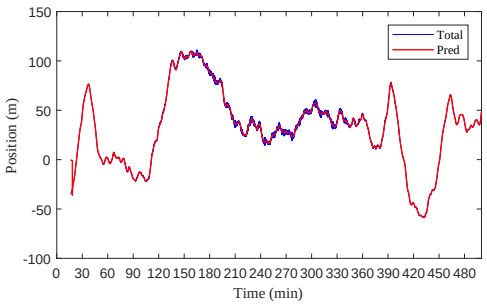
5.2.3 One DOF LF-Motion Prediction Without Excitation by the Actuators

The next simulation study indicates the behavior of the LF-motion prediction algorithm when the control input becomes zero. This is performed by smoothly reducing the gain of the control input to zero over 500 s. The control input is $\tau_c = 0$ for all scenarios at $6\,000\text{ s} < t < 15\,500\text{ s}$. This simulation indicates whether the NN can retain a learned model even when the system experiences no excitation.

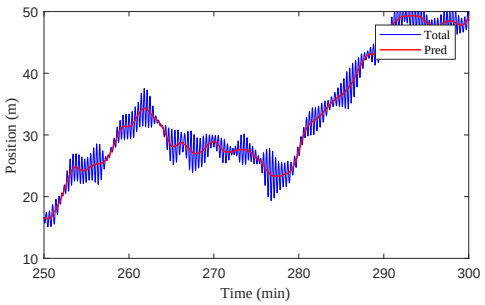
Figure 5.6 shows that the NN is able to reduce the oscillations. However, an increase of the error at around $t \approx 260$ min is noticeable, which could indicate that some undesired behavior has been learned and some important dynamics have been forgotten. Another possibility is that the NN has adapted to the specific range of the input values.

A method of locking weights is used to prevent this. Locking weights refers to prohibiting weights to update during training. The locked weights are therefore kept constant from the time of the locking. The idea is that weights, which are capable of modeling the system, are locked before the system experiences zero excitation by the actuators, which in this case is for $t \geq 6\,000$ s. By locking these weights, it is assumed that the disturbance can be modeled by the remaining weights, especially the weights corresponding to $y_\mathrm{p}(t)$.

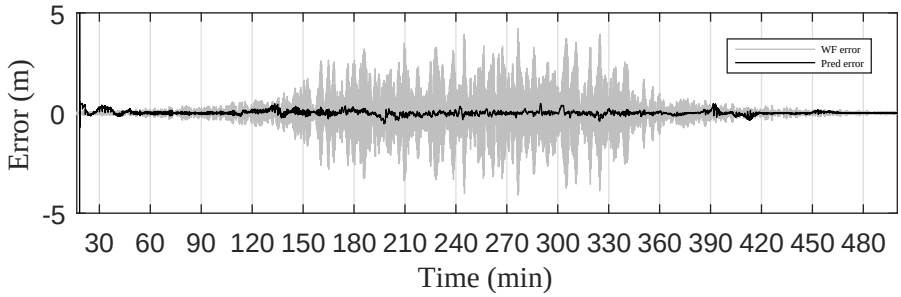
The STD for simulations locking different combinations of weights can be seen in Figure 5.7. The results in Figure 5.7a and 5.7b, where no weights and the weights for $u_\mathrm{p}(t)$, respectively, had been locked, should produce approximately the same result. The reason is that the training algorithm of the NN, which updates the weights proportional to the input. Hence, an input of zero leads



(a) Total motion and LF prediction.



(b) Enlarged image of the total motion and LF prediction.



(c) Prediction error compared to the oscillating disturbance. The MRMSE of the prediction is MRMSE = 0.1468

Figure 5.4: One DOF simulation: The algorithm is able to predict the LF-motion.

to no change of the corresponding weight. Comparing Figures 5.7a and 5.7b to Figures 5.7c and 5.7d demonstrates a similar behavior of all four scenarios. This indicates that the weights of $y_{\mathrm{p}}(t)$ are sufficient to model the changing behavior of the wave disturbance. On the other hand, large changes of the error at $t \approx 250$ min also indicate that the dynamics of the system are not entirely described by the locked weights for $u_{\mathrm{p}}(t)$, $\hat{y}_{LF,\mathrm{p}}(t)$ or layer-to-output. The fact that the STD for Figures 5.7c and 5.7d are greater towards the end, compared to Figures 5.7a and 5.7b could indicate that the NN has not learned the correct model of the system at the time of the locking.

5.2.4 One DOF LF-Motion Prediction Under Feedback Control

This simulation demonstrates the behavior of the LF predictor when the WF disturbance enters the feedback loop and is noticeable in the control input. These simulations are achieved by routing back the output of the system and thereby closing the loop. The error of the position from a reference position is calculated and fed into a simple proportional controller with a static gain of $K_{\mathrm{P}} = 0.5$. The reference position is set to be a scaled version of the control input in Section 5.2.3, which leads to a time interval, where the reference is zero. An important detail is that even though the system is in closed loop, the predictor itself still operates outside of the loop.

Figure 5.8 and Figure 5.9a demonstrate that the NN has problems converging towards a model when the reference is zero and the amplitude of the disturbance

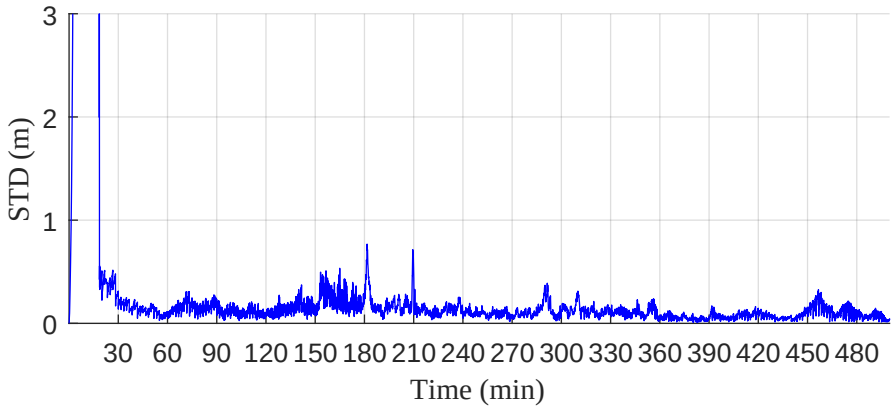
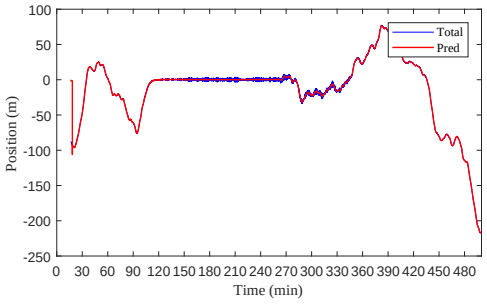
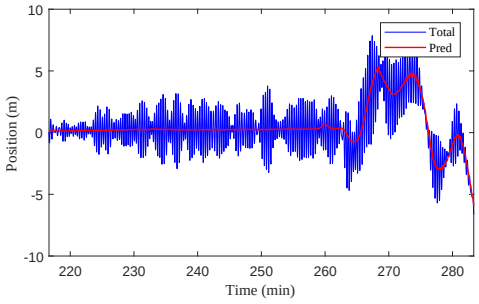


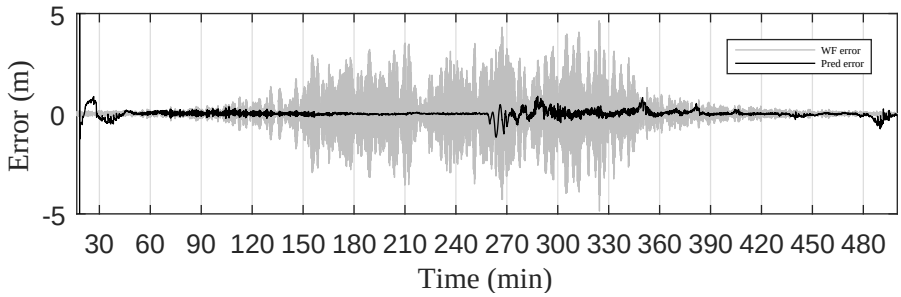
Figure 5.5: STD of the prediction error for nine one DOF simulations when the weights are updated.



(a) Total motion and LF prediction.

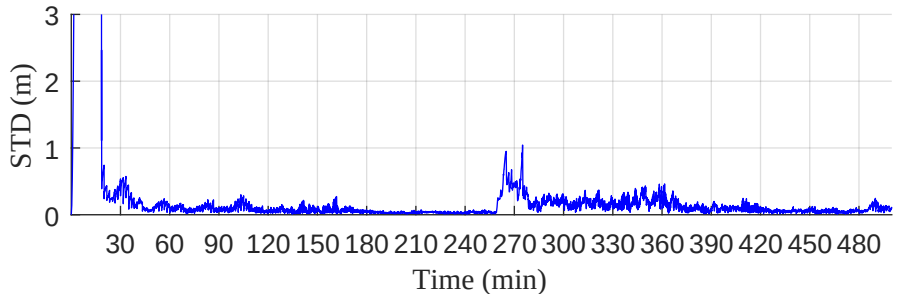


(b) Enlarged image of the total motion and LF prediction.

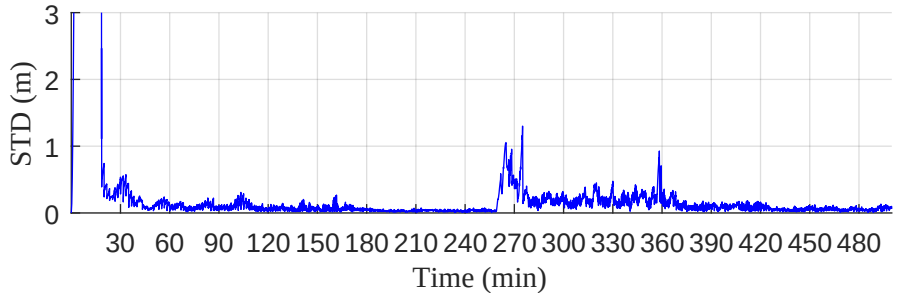


(c) Prediction error compared to the oscillating disturbance.

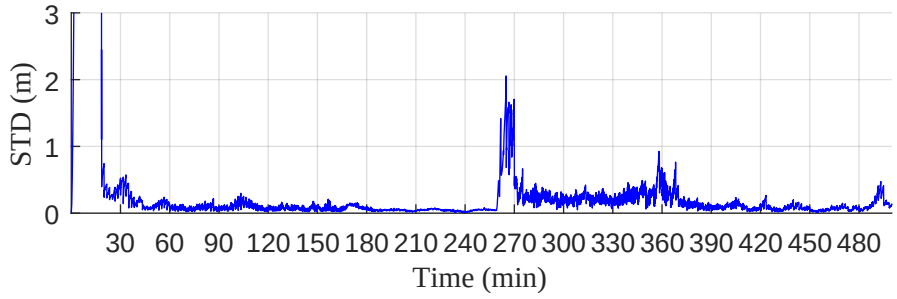
Figure 5.6: One DOF simulation without excitation by the actuators: The algorithm is able to predict the LF motion when $\tau_c = 0$, but a increase of the error is noticed when the system is excited by the actuators again.



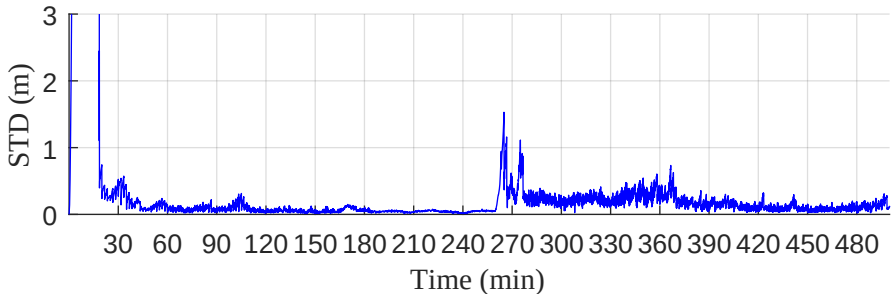
(a) Free weights.



(b) Weights for $u_{\mathrm{p}}(\tau)$ locked.

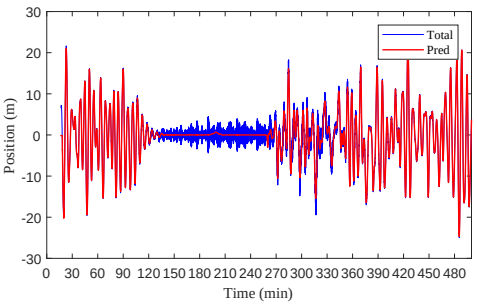


(c) Weights for $u_{\mathrm{p}}(\tau)$ and $\hat{y}_{LF,\mathrm{p}}(\tau)$ locked.

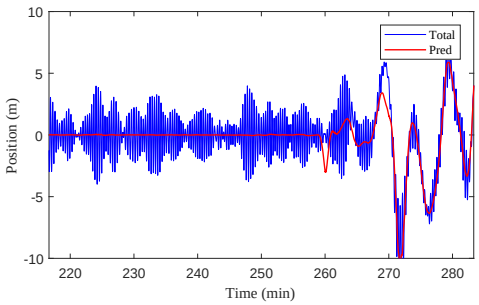


(d) Weights for $u_{\mathrm{p}}(\tau)$, $\hat{y}_{LF,\mathrm{p}}(\tau)$, and layer-to-output locked.

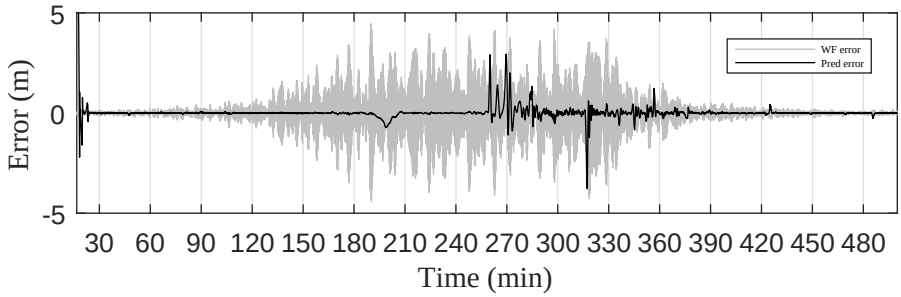
Figure 5.7: STD of the prediction error for nine one DOF open-loop simulations with zero excitation and locked weights.



(a) Total motion and LF prediction.



(b) Enlarged image of the total motion and LF prediction when the system is excited again.



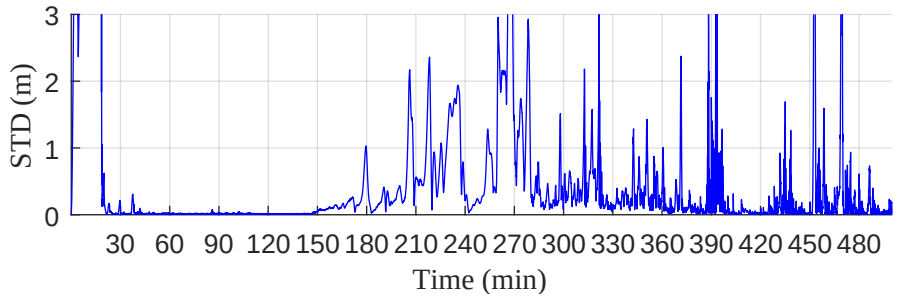
(c) Prediction error compared to the oscillating disturbance.

Figure 5.8: One DOF simulation under feedback control: An increase of the error is noticeable while the reference is zero and when the system is excited again.

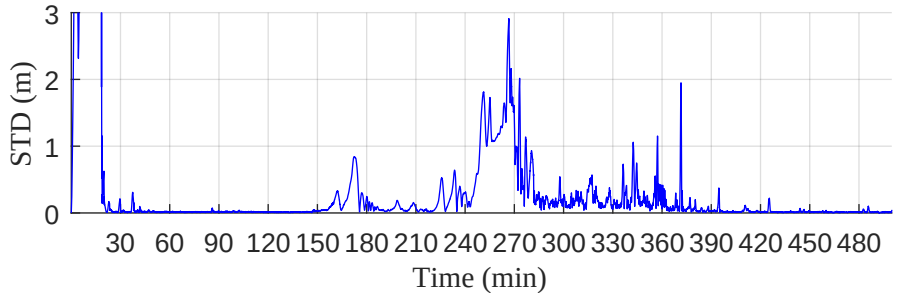
in the control input is large. Furthermore, when the reference changes again at $t = 15\,500$ s, the deviation from the actual LF motion is greater than seen in the open-loop simulation, see Figure 5.6. To be able to compare the closed-loop performance with the open-loop performance, the same weights where locked. The resulting STDs can be seen in Figure 5.9. The results demonstrate that even though the system gets excited the entire time, an increase of the error is noticeable at around $t \approx 250$ min. This could imply that the NN forgets the general behavior of the system due to training on a constant narrow interval.

Another interesting result is the divergence from the actual LF motion in case of free weights and a reference of zero. The reason for these results could be that even though the reference is zero, the feedback cause the control input to oscillate. That leads to correlating disturbance in the control input and the measured position. The correlating oscillation of the control input and measured position could prevent the NN to learn the behavior of the system.

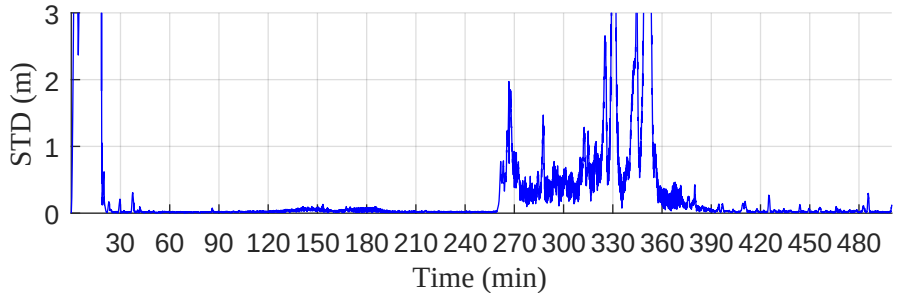
The advantage of locking weights is also demonstrated in Figure 5.9. Comparing Figures 5.9a - 5.9d, the assumption could be made, that keeping the weights responsible for the system model locked can improve the prediction when the reference is zero and the amplitude of the disturbance increases. On the other hand, locking the weights for $u_{\mathrm{p}}(t)$, $\hat{y}_{LF,\mathrm{p}}(t)$, and layer-to-output shows a behavior similar to Figure 5.7d. The result could be interpreted as the NN not having learned the correct model before locking the weights. This implies that locking the weights also prevents the predictor from improving.



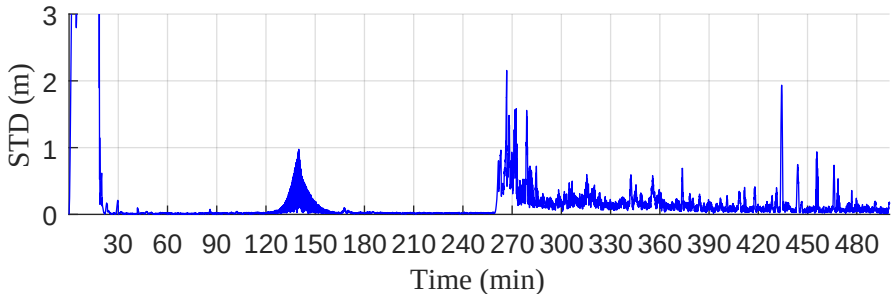
(a) Free weights.



(b) Weights for $u_{\mathrm{p}}(\tau)$ locked.



(c) Weights for $u_{\mathrm{p}}(\tau)$ and $\hat{y}_{LF,\mathrm{p}}(\tau)$ locked.



(d) Weights for $u_{\mathrm{p}}(\tau)$, $\hat{y}_{LF,\mathrm{p}}(\tau)$, and layer-to-output locked.

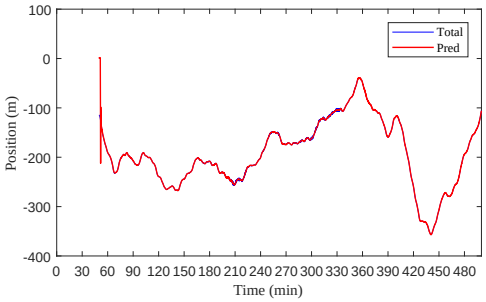
Figure 5.9: STD of the prediction error for nine one DOF simulations under feedback control and locked weights.

5.3 Three DOFs LF-Motion Prediction in Open Loop

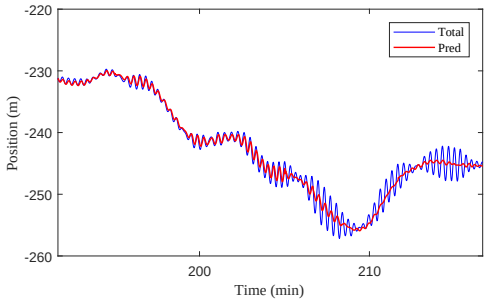
Evaluating the LF motion predictor on a more complex system with three inputs and three outputs, see Section 5.1.3, demonstrates whether the LF predictor can be extended to a three DOFs system or not. Furthermore, the nonlinearity is increased by adding the rotation matrix $\boldsymbol{J}_{3\text{DOF}}(\psi)$. The disturbance on the DOFs does not only change in frequency and amplitude, but also due to the heading relative to the disturbance. Apart from evaluating the possibility to extend the LF predictor to three DOFs, simulations on heading dependent disturbance show how well the LF predictor can adapt to the change. Because of the complexity of the system, the length of the training window is extended to $N = 3000$.

The results show that an extension to a three DOFs system is possible but that the quality of the prediction is decreased. Figure 5.10 depict the total motion of the vessel and the LF prediction in the north-position n, east position e, and heading relative to north ψ, respectively. Comparing these results, it seems that the algorithm is struggling to suppress the disturbance. Even though the high-amplitude WF disturbance is reduced, a prediction error greater than the oscillating disturbance is noticed when the disturbance amplitude is low. This can be noticed in Figure 5.11 at $330 \text{ min} < t < 430 \text{ min}$.

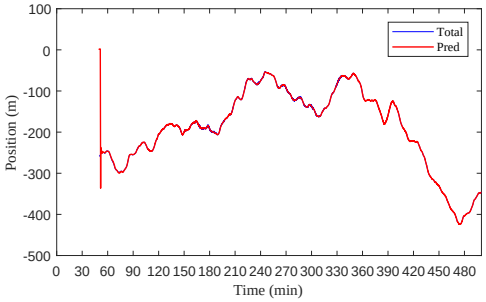
The results also show that the NN needs more time to adapt to the disturbance changing due to heading, dominating frequency, and amplitude. One example can be seen in Figure 5.10b at $200 \text{ min} < t < 215 \text{ min}$. It seems that the LF-motion predictor is able to predict the heading dependent disturbance to some extent, as long as the amplitude and frequency does not change. Figure 5.12, displaying the STD for nine runs, strengthens this assumption. The STD increases in the time intervals where the frequency of the disturbance changes and the amplitude of the disturbance first increases and then decreases fast which is at around $125 \text{ min} < t < 180 \text{ min}$ and $330 \text{ min} < t < 450 \text{ min}$, respectively. The summation of this simulation is, that it seems difficult to learn a model where two nonlinear rotation matrices $\boldsymbol{J}_{3\text{DOF}}(\psi)$ and $\boldsymbol{G}(\psi)$ are heading dependent and the frequency and the amplitude of the disturbance changes.



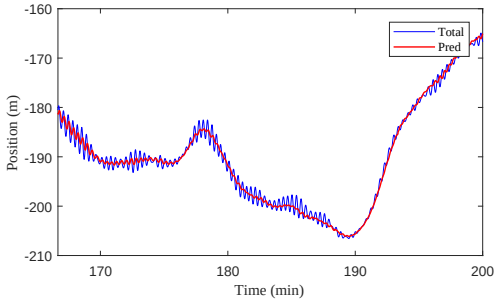
(a) Total motion and prediction of the north position.



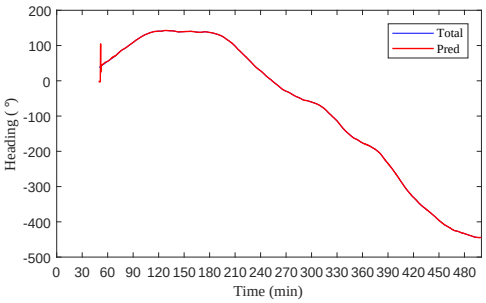
(b) Enlarged view of the total motion and prediction of the north position.



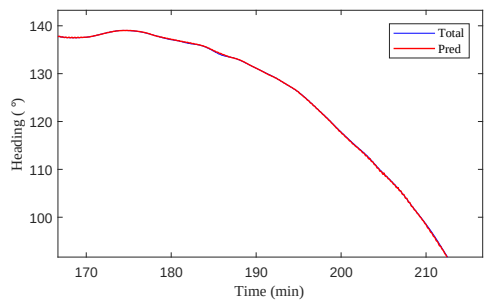
(c) Total motion and prediction of the east position.



(d) Enlarged view of the total motion and prediction of the east position.

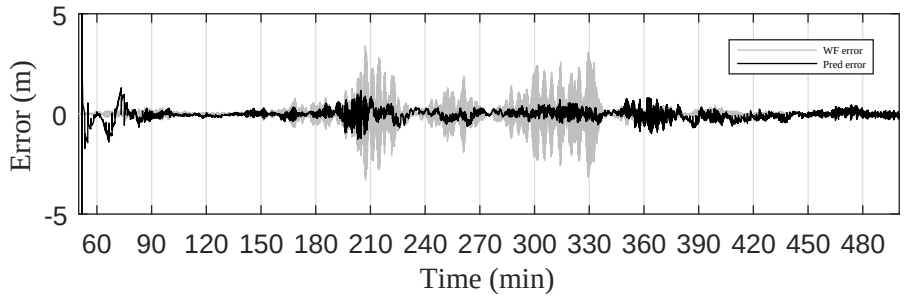


(e) Total motion and prediction of the heading.

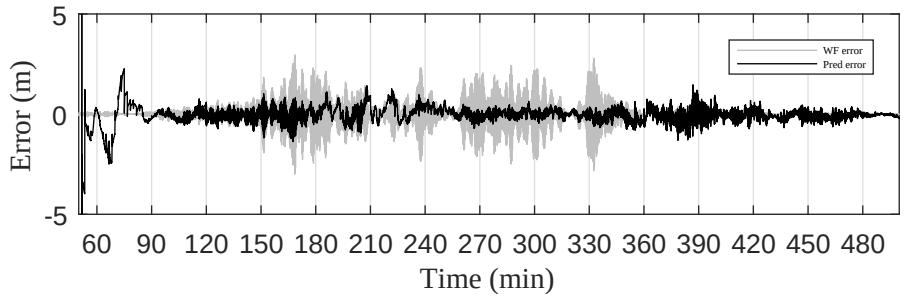


(f) Enlarged view of the total motion and prediction of the heading.

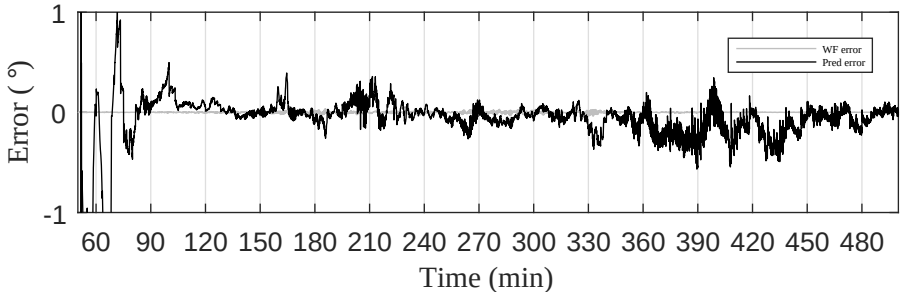
Figure 5.10: Three DOFs open-loop simulation. The total motion and the LF-motion prediction of the generalized position in the Earth-fixed frame are presented.



(a) Prediction error compared to the oscillating disturbance of the north position.

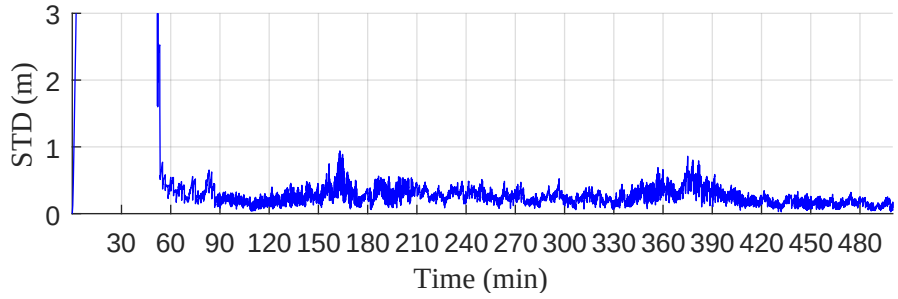


(b) Prediction error compared to the oscillating disturbance of the east position.

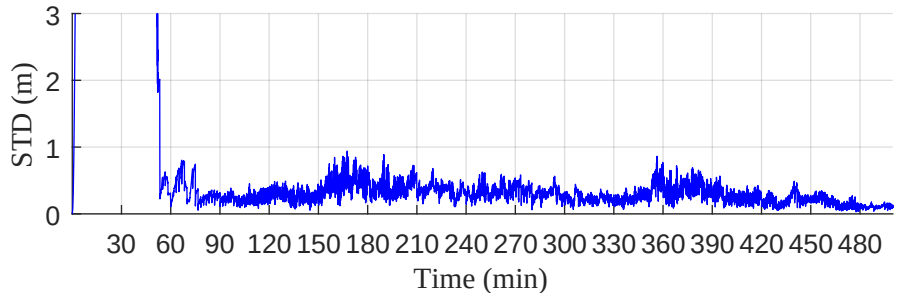


(c) Prediction error compared to the oscillating disturbance of the heading.

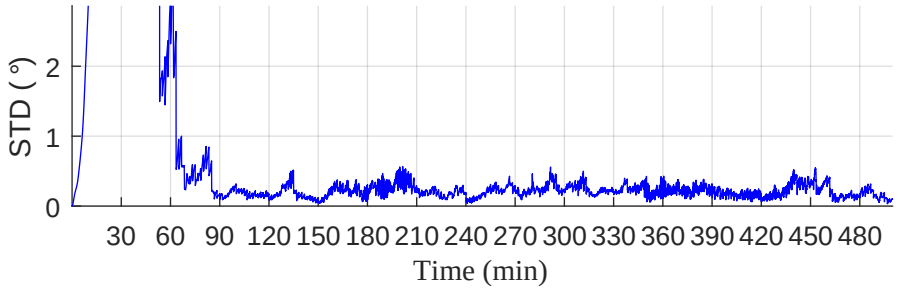
Figure 5.11: Three DOFs open-loop simulation. The prediction error of the LF-motion prediction of the generalized position in the Earth-fixed frame are presented.



(a) STD for the north position.



(b) STD for the east position.



(c) STD for the heading.

Figure 5.12: STD of the prediction error for nine open-loop three DOF simulations for the generalized position.

5.4 Ship Model Simulation

After evaluating the LF predictor on the simplified systems, it is evaluated on data from a ship simulator. Due to lack of time, these simulation results are initial results and need further investigation. Nevertheless, some early conclusions are made. The simulation model is based on the S175 container ship in the Marine Systems Simulator toolbox Fossen and Perez (2004). The principal quantities of the ship, linear hydrodynamic damping and frequency independent added mass are all from the MSS toolbox. The nonlinear damping, according to Fossen (2011, Sec. 6.4), is calculated based on principal quantities of the ship. Regarding the modeling of the disturbances, the current and wind disturbance models are based on Fossen (2011, Sec. 8.1 and 8.3), while the wave disturbance is based on force response amplitude operators and the wave spectrum from the MSS toolbox. The wind and current disturbances are not enabled in these simulations. The

dominating frequency of the wave disturbance is 0.9 rad/s.

The simulation applies DP using PID control on the generalized position (n, e, ψ) in three DOFs. This is achieved by routing back the measured generalized position and calculating the error from the reference. The reference is zero for all DOFs. The error is translated into the body-fixed frame. Three decoupled PID controllers adjust force on surge, sway and yaw according to the error. The gains are listed in Table A.6.

5.4.1 LF-Motion Prediction in Closed Loop

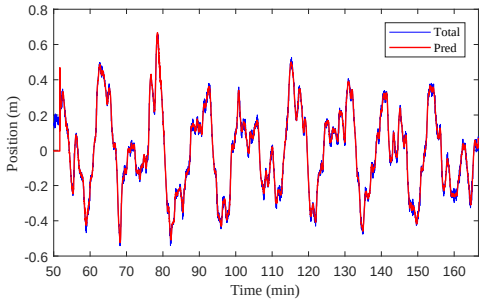
Running the LF motion predictor on data from a model of a ship illustrates the performance of the algorithm in the application it is intended for. Even thought the DP runs in closed loop, the algorithm is again used passively outside of the loop. As in Section 5.3, the outputs are given in the Earth-fixed reference frame. The number of delayed past prediction inputs to the RNN is decreased to $n_{\text{fb}} = 2$.

Figure 5.13 depict the total motion of the vessel and the LF prediction in the north position n, east position e, and heading relative to north ψ, respectively. Figure 5.14 presents the corresponding prediction errors compared to the oscillation of the wave. The results demonstrate that the algorithm is able to learn the model to predict the LF motion, except for the noticeable increase of the prediction error at $t = 80$ min.

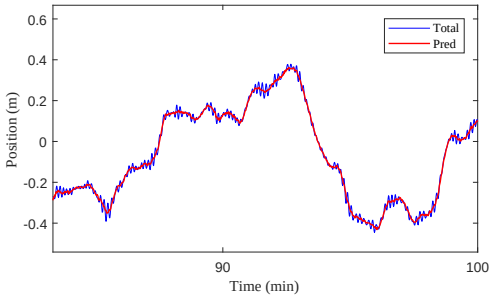
It seems that the results shown in Figure 5.13 are better than the results gained in Section 5.3. There could be some advantages using this algorithm for position keeping rather than just predicting the LF motion of a cruising ship. Trying to keep a constant heading leads to an almost constant rotation matrix $\boldsymbol{J}(\phi, \theta, \psi)$ and almost constant wave disturbance on the DOFs. Furthermore, the frequency of the wave disturbance does not change during these simulations due to limitations in the simulink model.

Using this algorithm in DP, and thereby trying to keep a certain position, also favors the algorithm in another way. This is due to the scaling used to preprocess the data. During position keeping, the position has well defined bounds and the scaling factor can be minimized. Scaling has the disadvantage that a large scaling factor compared to the WF motion will drive the observed disturbance to zero. This implies that the oscillation disturbance relative to the total range of motion becomes small. This is not the case in DP, where the motion of the disturbance relative to the total range of motion remains large. That could be an additional factor that explains the rather satisfactory results of this simulation.

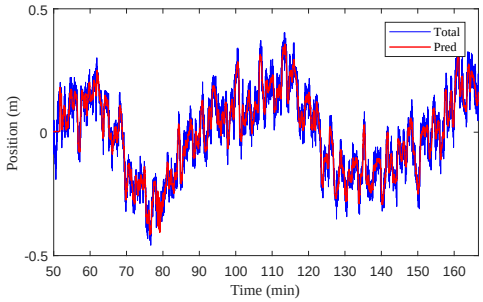
The increase of the prediction error at $t = 80$ min, which is seen in Figure 5.15 as an increase of the STD to around 0.3 m in sway, 0.1 m in sway, and 0.5 deg in heading, could be the result of one of the problems within this algorithm. In the current state, it is not possible to ensure a stable and always converging algorithm. It is only possible to increase the probability by, for example, creating accurate target data, choosing the training data set N large enough, choosing the correct regressors and NN hyperparameters, and tuning the regularization parameter. Therefore, sudden and large deviations from the LF motion can appear at any time.



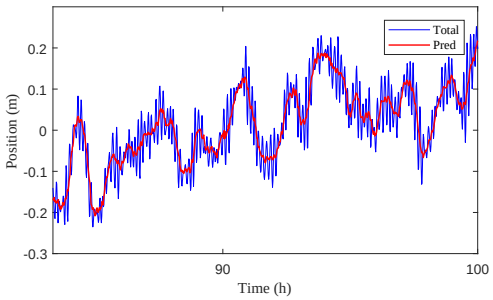
(a) Total motion and LF prediction of the north position.



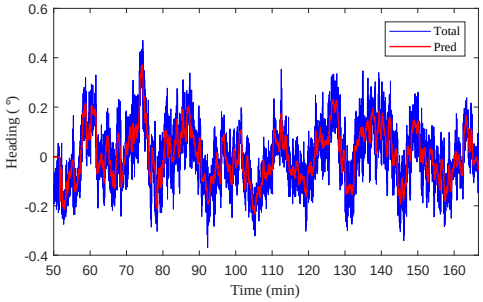
(b) Enlarged image of the total motion and LF prediction of the north position



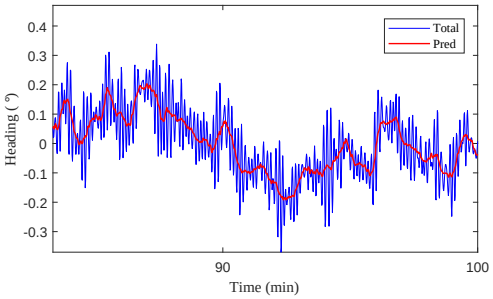
(c) Total motion and LF prediction of the east position.



(d) Enlarged image of the total motion and LF prediction of the east position.

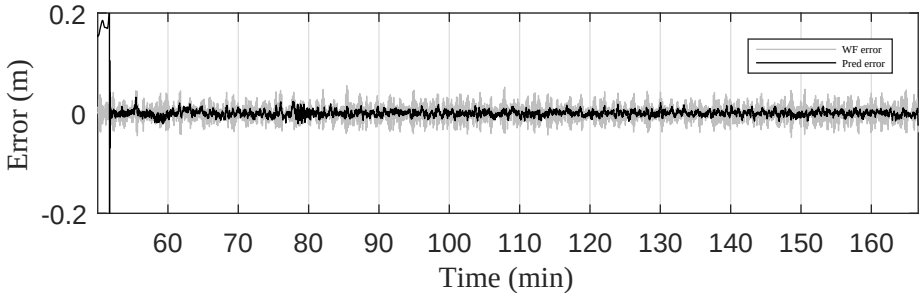


(e) Total motion and LF prediction of the heading.

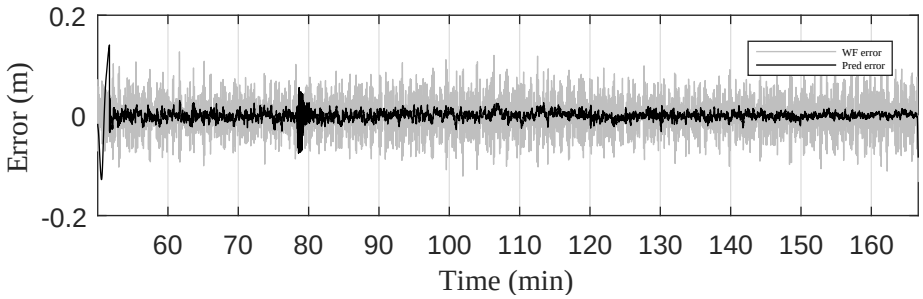


(f) Enlarged image of the total motion and LF prediction of the heading.

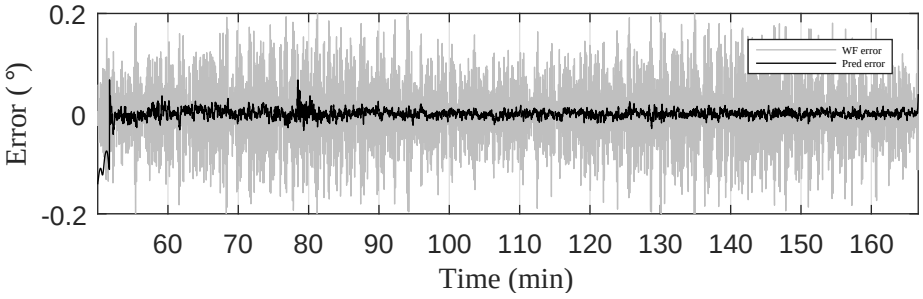
Figure 5.13: The generalized position in the Earth-fixed frame of a closed-loop vessel simulation: The LF predictor is able to reduce the oscillations.



(a) Prediction error of the north position compared to the oscillating disturbance.

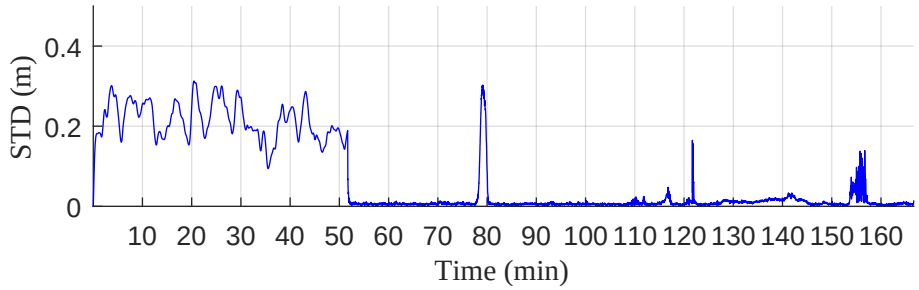


(b) Prediction error of the east position compared to the oscillating disturbance.

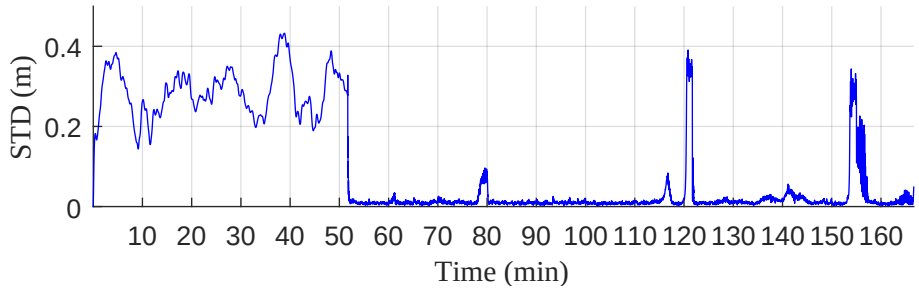


(c) Prediction error of the heading compared to the oscillating disturbance.

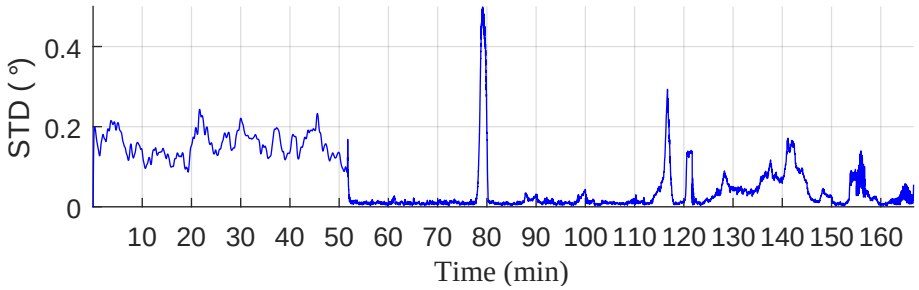
Figure 5.14: The prediction error of the generalized position in the Earth-fixed frame of a closed-loop vessel simulation.



(a) STD for the north position.



(b) STD for the east position.



(c) STD for the heading.

Figure 5.15: STD of the prediction error for the generalized positions for nine closed-loop vessel simulations.

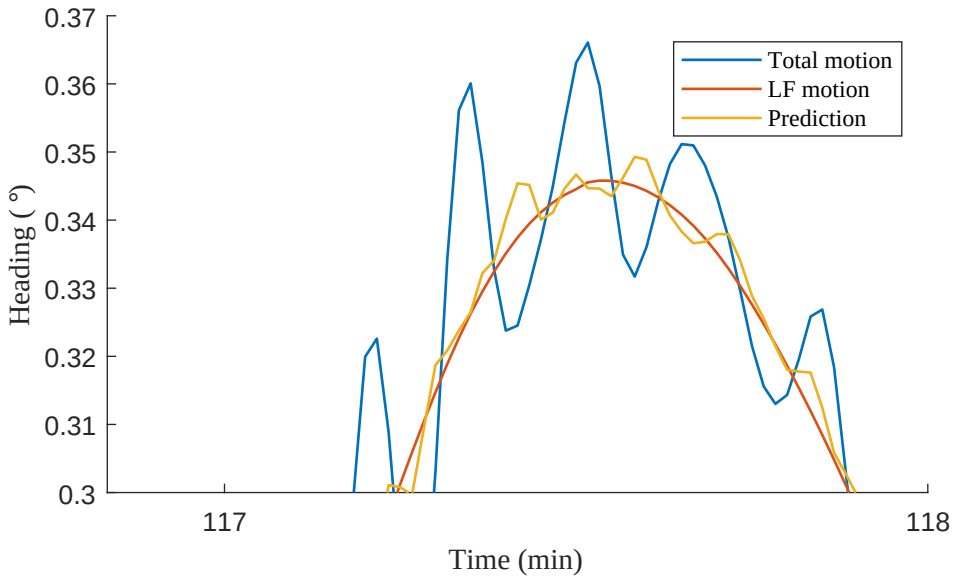


Figure 5.16: Enlargement of the total heading, the LF heading and the predicted LF heading. Some oscillating motion occur in the prediction in the opposite direction to the actual motion of the vessel.

5.4.2 Comparison with Nonlinear Observer

After evaluating the LF predictor outside the closed loop, the predictor is evaluated inside the closed loop. These simulation show if using the LF predictor actually decreases the oscillation in the control signal. For the reason that no simulink implementation for the proposed prediction algorithm yet exists, the implementation of the LF predictor is done by using the NN, which was trained during the simulations shown in Section 5.4.1, as a static predictor. The input to the NN is the control signal and the measured position. The output of the NN is routed to a controller. The reference position has the values zero for all DOFs, however, only the north position, east position, and heading are controlled by the controller. Similar to the simulation in Section 5.4.1, the frequency of the disturbance is not varied over time.

Controlling the system with the PID controller used in Section 5.4.1 leads to an unstable system. The reason could be that despite suppressing the WF motion, the LF prediction is noisy and oscillations can occur, see Figure 5.16. The graph depicts that even though the LF position is predicted satisfactory, the derivative of the LF-motion prediction can get a sign opposite to the derivative of the actual total motion. This will cause the derivative term of the PID controller to steer the ship with the direction of the wave. The reason for the occurring oscillations in the prediction could be the loss function used while training. Training on the MSEREG of the LF motion and the prediction does not take the sign of the derivative into account. This implies that an small oscillating error with a derivative with the same sign as the total motion gives the same error as a small oscillating error with a derivative of the opposite sign. Even tough the error is the same, this behavior is unwanted. To counter the instability problem, a PI controller is chosen for this simulation. The gains of the PI controller can be

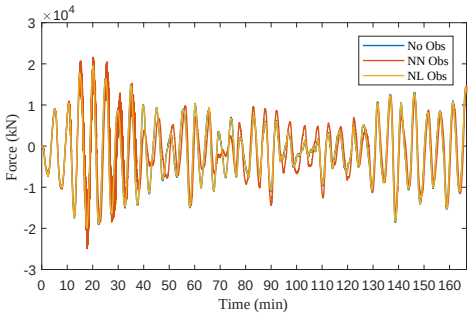
found in Table A.7.

Running the static NN inside the closed loop enables a comparison to a model-based nonlinear observer. The chosen nonlinear observer is derived in Fossen and Strand (1999). The observer presupposes a linear model of the vessel. The numerical values of the tuning parameters are listed in Table A.5. For comparison, three simulation with identical wave disturbance are executed and compared. The first simulation does not include an observer in the closed loop and is therefore a reference simulation. The second and third simulation uses the nonlinear observer and the static NN, respectively, inside the closed loop to estimate the LF motion. Including an observer into the feedback loop should reduce the high-frequency oscillations of the force applied by the actuators.

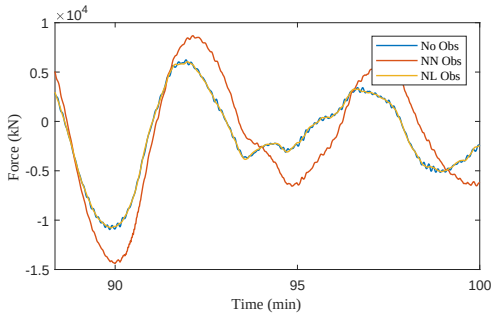
Figure 5.17 demonstrates the forces applied by the actuators to control the vessel. In Figures 5.17a, 5.17c, and 5.17e a high-frequency oscillation with great amplitude is noticeable in all DOFs from 10 min to 40 min. The oscillating behavior arieses when the static NN is used. Figures 5.17a and 5.17c are enlarged views of the control signal in sway and yaw. They demonstrate that the oscillation of the control signal due to waves reduced at a later time when using the static NN as LF-motion predictor. The prediction is close to the prediction of the nonlinear observer. Nevertheless, one can also notice sudden and unexpected change of the control signal, which do not occur when using the nonlinear observer. Figure 5.18 presents the generalized position in the Earth-fixed frame corresponding to the control inputs demonstrated in Figure 5.17. One can see that the positions are almost identical when no observer and the nonlinear observer is used. The graphs also demonstrate that, when using the static NN, the position deviation is larger.

Some behavior of the control input in Figure 5.17 can be explained with the position seen in Figure 5.18 and vice versa. The mismatch of the control signals seen in Figure 5.17b can be explained by the mismatch of the position in Figure 5.18a. A mismatching position will lead to a mismatch of the control signals.

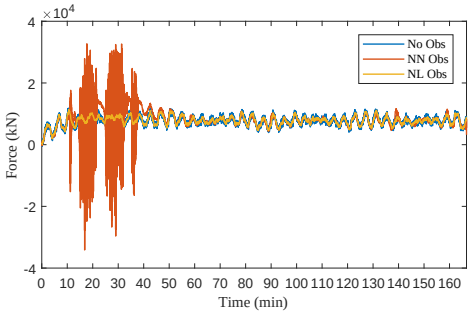
That the position deviates in all DOFs, when using the static NN, is a result of the heavy oscillation of the control signal from 10 min to 40 min. This oscillation, on the other hand, is caused by false LF-position predictions by the NN. The circumstances while training the NN can be a reason for the false prediction. The neural network was training on the data presented in Figure 5.13, which was scaled into the range -1 to 1. Using the same NN with the same scaling will result in inputs to the NN that are greater than the inputs used while training. These larger inputs can lead to saturation of the activation function which in turn lead to erroneous prediction.



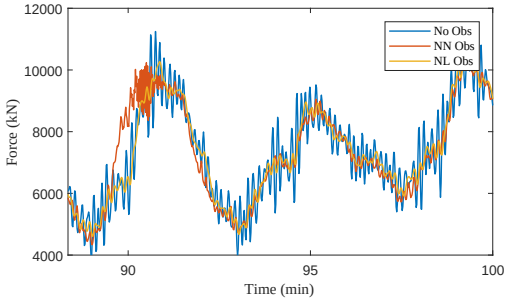
(a) Applied actuator force in surge.



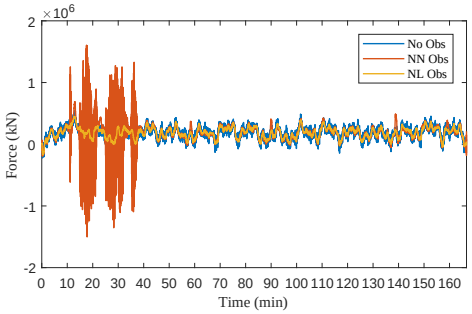
(b) Enlarged view of the applied actuator force in surge.



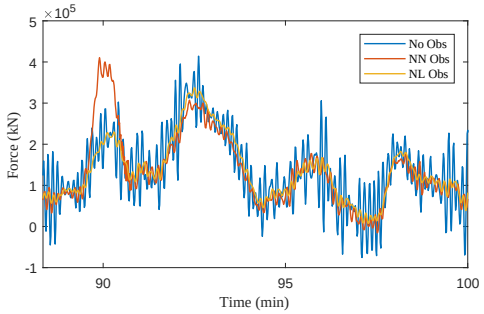
(c) Applied actuator force in sway.



(d) Enlarged view of the applied actuator force in sway.

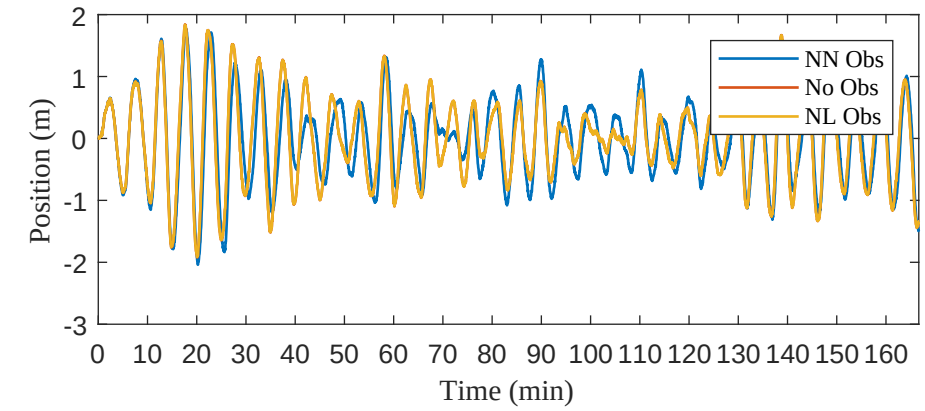


(e) Applied actuator force in yaw.

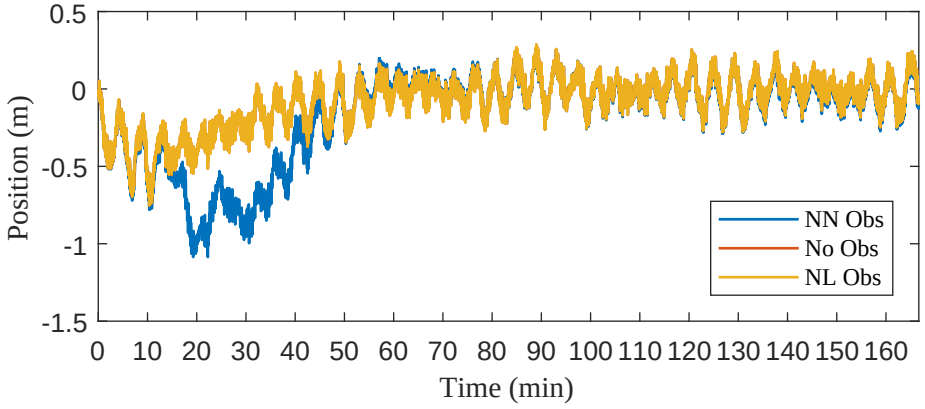


(f) Enlarged view of the applied actuator force in yaw.

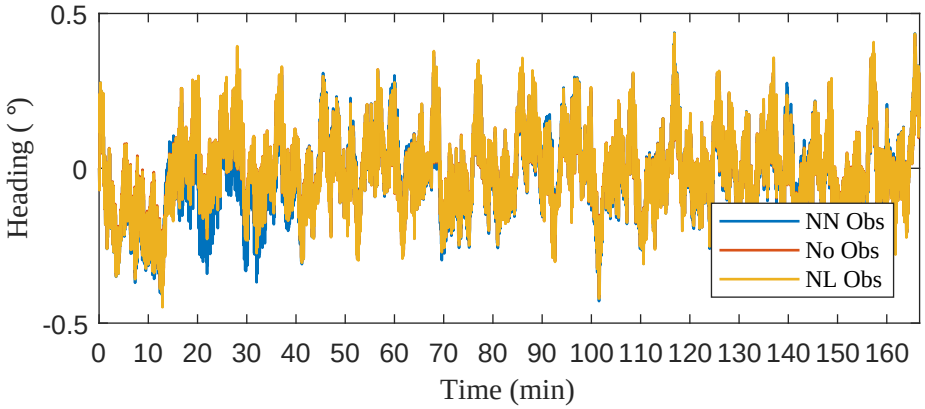
Figure 5.17: Comparison of the affect on the control input when using no observer at all (No Obs), a nonlinear model-based observer (NL Obs), and a static LF predictor (NN Obs). The graphs illustrate the applied force by the actuators in surge, sway, and yaw.



(a) The north position of the vessel.



(b) The east position of the vessel.



(c) The heading of the vessel.

Figure 5.18: Comparison of the generalized Earth-fixed positions when using no observer at all (No Obs), a nonlinear model-based observer (NL Obs), and a static LF predictor (NN Obs).

6 Conclusions and Future Work

In this thesis a predictor was proposed to predict the low-frequency motion of a system affected by an oscillating wave disturbance with changing frequency and amplitude. The predictor is able to predict the low-frequency motion with minimal knowledge about the system. The proposed predictor is based on a recurrent neural network. The recurrent neural network is trained by supervised learning to predict the low-frequency motion of a system. The supervised learning uses back-dated data to learn the dynamics of the system. Furthermore, to enable training, the target data, which is needed for supervised learning, is created from that back-dated data by signal processing.

The results show that it is in fact possible to predict the low-frequency motion of a system affected by oscillatory disturbances using the proposed predictor. The predictor can learn the necessary information from the available data. However, the main goal is just partly reached. The results show that the predictor can learn how to predict the low-frequency motion but it can not be guaranteed that the neural network converges or the system is learned correctly using the implemented training method. Moreover, the predicted low-frequency motion is not guaranteed to be free from oscillations.

Interesting information was gained from the simulations. To favor the learning and avoid forgetting important dynamics, the system needs to be excited continuously. If the system is not excited enough, it is possible to lock weights of the neural network to keep the learned dynamics. Furthermore, it is sufficient to update the weights between the position input and the layer, together with the biases, to adapt to changing disturbances. The predictor can easily be extended to three degrees of freedom and is to some extent able to predict the low-frequency motion but struggles to learn rotational dependencies. Initial evaluation of the predictor on a ship model applying dynamic positioning show promising results. These results indicate that the predictor can reduce the oscillating behavior of the control signal, if the ship tries to keep a constant position and moves inside a limited range.

The simulation results show that it is possible to learn wave filtering using machine learning and it is worth to pursue this further. The information needed to predict the low-frequency position is available in the position data and the method of using a back-dated data for learning is promising.

An interesting next step would be to predict not only the position but also the velocity. The target for a velocity predicting neural network could be approximated as the derivative of the position target. Being able to predict the velocity would enable more advanced control.

Due to the lack of time the investigation of the predictor on the ship model was limited. Therefore, the behavior of the predictor in this model should be evaluated further and more thoroughly. Furthermore, the predictor should be implemented into simulink to enable continuous use together with the ship model.

A lot was learned during this study and several areas of the predictor have to

be improved in future work. The biggest drawback is the uncontrolled training of the predictor, which does not ensure correct learning or convergence. In further studies, the training has to be made robust and controlled to encourage the use of neural networks. Furthermore, the loss function should be improved to control the sign of the derivative of remaining oscillations in the predicted motion.

Another hurdle to overcome is the preprocessing needed for neural networks. The preprocessing limits the use of the predictor to a known position range. This limitation is not a problem in position keeping, where certain position boundaries are known, but complicates, for example, cruising.

Mentioning the difficulties above, it is clear that model-based observers are, at the moment, superior to this approach. Anyhow, the thesis shows that collected data contains valuable information, which could be utilized.

7 Svensk sammanfattning

Fartyg påverkas konstant av olika miljöbetingade krafter. Dessa kan delas in i lågfrekventa och högfrekventa krafter. Till de lågfrekventa krafterna räknas vind och havsströmmar. Vågor ger upphov till både drift, en lågfrekvent störning, och en oscillerande kraft, en högfrekvent störning. De oscillerande krafterna kallas vågfrekvenskrafter (eng. wave-frequency forces). Det är de oscillerande vågfrekvenskrafterna som leder till att fartyget svänger fram och tillbaka med vågorna. De lågfrekventa och högfrekventa krafterna inverkar på och utgör på olika sätt störningar för fartygets position. Till exempel vid dynamisk positionering leder de lågfrekventa krafterna till att fartyget driver från den önskade positionen. Då en sådan drift är oönskad krävs det att styrsystemet kan kompensera för den oönskade driften. Vågfrekvenskrafterna leder å andra sidan till att fartygets position pendlar kring den önskade positionen. Vid dynamisk positionering är denna begränsande avvikelse tolererbar och kräver ingen kompensation av styrsystemet. Dock kommer styrsystemet att försöka kompensera för varenda våg ifall inte vågfiltrering används. Det leder till en oscillerande rörelse i fartygens ställdon vars frekvens är för hög för att uppnå någon märkbar skillnad i positionen. Den här onödiga rörelsen sliter på mekaniken och leder till en ökad bränsleförbrukning. Detta beteende kan motverkas genom vågfiltrering av positions- och hastighetsdata. Med hjälp av vågfiltrering vill man bearbeta den uppmätta datan. Man vill behålla informationen om fartygets lågfrekventa rörelser samtidigt som man vill filtrera bort de oscillerande vågstörningarna. Vågfiltrering utförs nuförtiden genom att applicera bandspärr- och lågpassfilter eller genom att använda modellbaserade tillståndsobservatörer, så som exempelvis Kalmanfilter. Nackdelarna med dessa metoder är att de skapar fasförskjutning eller att de är starkt beroende av att man jobbar med en precis modell av fartyget och störningarna.

7.1 Metod och resultat

I den här studien undersöktes ifall det är möjligt att lära en algoritm att prediktera den lågfrekventa rörelsen av ett fartyg, det vill säga fartygets rörelse utan oscillationer, utan att ha exakt kunskap om fartygets eller de rådande störningarnas dynamik. För att uppnå detta användes återkopplade neuronnät (eng. recurrent neural networks) som maskininlärnings-metod (eng. machine learning). Metoden som presenterades i arbetet är en prediktor som använder uppmätt data om fartygets position för att förutsäga fartygets lågfrekventa rörelse. Prediktorn utgörs av ett återkopplat neuronnät. För att lära neuronnätet att prediktera den lågfrekventa rörelsen måste det först tränas med hjälp av så kallat träningsdata. Nätet tränades genom övervakad träning vilket betyder att det tränades så att en viss insignal ska ge en förutbestämd utsignal. Utsignalen som nätet skulle återskapa, måldatan (eng. target data), skapades innan träningen. För prediktorn som redogjordes för i denna studie krävdes att måldatan presenterar fartygets lågfrekventa rörelse för de korresponderande insignalerna i träningsdatan. Mål-

datan skapades därför genom att uppskatta den aktuella vågfrekvensen och sedan applicera ett bandspärr- och lågpassfilter på den uppmätta positionsdatan. Vågstörningens frekvens estimerades genom att filtrera aktuell data med hjälp av ett högpassfilter och därmed eliminera fartygets frekvenskomponenter och bevara vågstörningens frekvenskomponenter. Därefter uppskattades signalens effektspektrum och vågfrekvensen kan bestämmas. Vidare användes den uppskattade frekvensen för att skapa ett bandspärrs- och lågpassfilter. Eftersom fasförskjutning vid filtrering är oönskad applicerades framåt-bakåt filtrering. De kaskadkopplade filtren dämpar den oscillerande vågrörelsen i den uppmätta positionsdatan och skapar därmed måldata som är fartygets lågfrekventa rörelse. Efter att måldatan hade genererats tränades det återkopplade neuronnätet på träningsdatan för att lära sig prediktera den lågfrekventa rörelsen utifrån den verkliga positionsmätningen. Den tränade prediktorn användes efter träningen för att prediktera den lågfrekventa rörelsen för att därmed minska på ställdonets oönskade oscillerande rörelser. Eftersom vågstörningens egenskaper förändras och det krävs mycket data för neuronnätet att lära sig en bra modell av fartyget och störningarna, tränades nätet regelbundet. Efter att prediktorn hade utvecklats, evaluerades denna med hjälp av simuleringar där tre olika system med ökande komplexitet användes. Det första systemet hade en frihetsgrad, vilket motsvarar rörelse i endast en riktning. Detta utökades sedan till ett system bestående av tre frihetsgrader, vilket motsvarar fartygets rörelse i det horisontella planet. Avslutningsvis utvärderades prediktorns prestation på en fartygsmodell. Under simuleringarnas gång ändrades frekvensen och amplituden av vågstörningen. Simuleringarna utfördes genom att simulera modellerna och sedan använda prediktorn på simuleringsdata. Vid simulering av systemet med en frihetsgrad kunde det fastställas att den insamlade mätdatan kan innehålla tillräckligt med information om störningen och fartygets dynamik för att lära prediktorn vågfiltrering. Det här gäller då systemet är väl exciterat. Då komplexiteten höjdes till ett system med tre frihetsgrader introducerades även en riktningsberoende vågstörning och en rotationsmatris som transformerar fartygets hastighet mellan fartygets och jordens referensram. Resultaten som presenterades i studien visar att det är möjligt att utöka prediktorn till flera frihetsgrader. Simuleringsresultaten visar även att prediktorn har svårigheter med att lära sig rotationsberoendet. Inledande simuleringar med en verklig fartygsmodell under dynamisk positionering indikerar att prediktorn för vågfiltrering som utvecklats inom ramen för denna studie är lovande och bör studeras vidare.

7.2 Sammanfattning

Sammanfattningsvis kan sägas att resultaten från den här studien visar att det är möjligt att genom maskininlärning utföra vågfiltrering, det vill säga lära en algoritm att prediktera den lågfrekventa rörelsen av ett fartyg. Den information som behövs för träning och inlärning finns i den uppmätta positiondatan och man bör fortsätta forska inom detta område. I framtida studier bör man sträva efter att utveckla träningsmetoden för neuronnätet. Ännu garanterar den inte konvergens och felaktiga prediktioner kan förekomma.

Bibliography

F. Abdollahi, H. A. Talebi, and R. V. Patel. A stable neural network-based observer with application to flexible-joint manipulators. *IEEE Transactions on Neural Networks*, 17(1):118–129, January 2006. doi: 10.1109/TNN.2005.863458.

Z. Aydogmus and O. Aydogmus. A comparison of artificial neural network and extended kalman filter based sensorless speed estimation. *Measurement*, 63: 152–158, March 1, 2015. doi: 10.1016/j.measurement.2014.12.010.

J. G. Balchen, N. A. Jenssen, E. Mathisen, and S. Saelid. Dynamic positioning of floating vessles based on kalman filtering and optimal control. In *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 852–864, Dec 1980. doi: 10.1109/CDC.1980.271924.

D. J. W. Belleter, R. Galeazzi, and T. I. Fossen. Experimental verification of a global exponential stable nonlinear wave encounter frequency estimator. *Ocean Engineering*, 97:48–56, March 15, 2015. doi: 10.1016/j.oceaneng.2014.12.030.

D. J. Belleter, D. A. Breu, T. I. Fossen, and H. Nijmeijer. A globally k-exponentially stable nonlinear observer for the wave encounter frequency. *IFAC Proceedings Volumes*, 46(33):209 – 214, 2013. ISSN 1474-6670. doi: 10.3182/20130918-4-JP-3022.00016. 9th IFAC Conference on Control Applications in Marine Systems.

S. Bennett. Nicholas minorsky and the automatic steering of ships. *IEEE Control Systems Magazine*, 4(4):10–15, November 1984. doi: 10.1109/MCS.1984.1104827.

B. J. Copeland. Artificial intelligence, Aug 2018. URL `https://www.britannica.com/technology/artificial-intelligence`. <Accessed: 2018-08-27>.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, /12/01 1989. doi: 10.1007/BF02551274.

F. M. Dias, A. Antunes, J. Vieira, and A. Mota. A sliding window solution for the on-line implementation of the levenbergmarquardt algorithm. *Engineering Applications of Artificial Intelligence*, 19(1):1 – 7, 2006. ISSN 0952-1976. doi: 10.1016/j.engappai.2005.03.005.

H. Enshaei and R. Birmingham. Monitoring of dynamic stability via ship's motion responses. In K. S. andN.Themelis andA.D.Papanikolaou, editor, *Proceedings of the 11th International Conference on the Stability of Ships and Ocean Vehicles*, pages 707–717, 2012.

F. D. Foresee and M. T. Hagan. Gaussnewton approximation to bayesian learning. In *Neural Networks,1997., International Conference on*, volume 3, pages 1930–1935 vol.3, Jun 1997. doi: 10.1109/ICNN.1997.614194.

T. I. Fossen and T. Perez. Marine systems simulator (mss), 2004. URL `http://www.marinecontrol.org`. <Accessed: 2018-07>.

T. I. Fossen and T. Perez. Kalman filtering for positioning and heading control of ships and offshore rigs. *IEEE Control Systems*, 29(6):32–46, December 2009. doi: 10.1109/MCS.2009.934408.

T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, Hoboken, 1. aufl. edition, 2011. ISBN 1119991498.

T. I. Fossen and J. P. Strand. Passive nonlinear observer design for ships using lyapunov methods: full-scale experiments with a supply vessel. *Automatica*, 35 (1):3–16, January 1, 1999. doi: 10.1016/S0005-1098(98)00121-6.

M. T. Hagan and M. B. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, Nov 1994. ISSN 1045-9227. doi: 10.1109/72.329697.

S. S. Haykin. *Neural networks and learning machines*. Pearson, Upper Saddle River [u.a.], 3. ed. edition, 2009. ISBN 0131293761.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35, 1960. doi: 10.1115/1.3662552.

P. Kim. *MATLAB Deep Learning : With Machine Learning, Neural Networks and Artificial Intelligence*. Apress L. P, Berkeley, CA, Jun 15, 2017. ISBN 9781484228449.

A. Krogh and J. Hertz. A simple weight decay can improve generalization. 4, 02 1992.

T. Lauvdal and T. I. Fossen. A globally stable adaptive ship autopilot with wave filter using only yaw angle measurements. *IFAC Proceedings Volumes*, 28(2): 262–269, May 1, 1995. doi: 10.1016/S1474-6670(17)51680-8.

K. P. Lindegaard and T. I. Fossen. A model based wave filter for surface vessels using position, velocity and partial acceleration feedback. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, volume 1, pages 946–951 vol.1, 2001. doi: 10.1109/.2001.980231.

H. Lohninger. *Teach/Me Data Analysis*. Springer-Verlag, Berlin-New York-Tokyo, 1999. ISBN 3-540-14743-8.

D. P. Mandic and J. Chambers. *Recurrent Neural Networks for Prediction: Learning Algorithms,Architectures and Stability*. John Wiley & Sons, Inc., New York, NY, USA, 2001. ISBN 0471495174.

N. Morgan and H. Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 630–637. Morgan-Kaufmann, 1990.

K. S. Narendra and K. Parthasarathy. Neural networks and dynamical systems. *International Journal of Approximate Reasoning*, 6(2):109 – 131, 1992. ISSN 0888-613X. doi: 10.1016/0888-613X(92)90014-Q.

Neural Network Toolbox, 2017. The MathWorks, Natick, MA, USA.

U. D. Nielsen. Estimations of on-site directional wave spectra from measured ship responses. *Marine Structures*, 19(1):33 – 69, 2006. ISSN 0951-8339. doi: 10.1016/j.marstruc.2006.06.001.

T. Perez. *Ship Motion Control: Course Keeping and Roll Stabilisation Using Rudder and Fins*. Springer-Verlag, London, 2005. ISBN 1-85233-959-4.

L. Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761 – 767, 1998. ISSN 0893-6080. doi: 10.1016/S0893-6080(98)00010-0.

T. Schei. Wave disturbance filtering in dynamic positioning systems. *IFAC Proceedings Volumes*, 28(2):27 – 34, 1995. ISSN 1474-6670. doi: 10.1016/S1474-6670(17)51647-X. 3rd IFAC Workshop on Control Applications in Marine Systems, Trondheim, Norway, 10-12 May.

H. Siegelmann, W. Horne, and C. Lee Giles. Computational capabilities of recurrent narx neural networks. 27:208–15, 02 1997.

Signal Processing Toolbox, 2017. The MathWorks, Natick, MA, USA.

J. Sjöberg, H. Hjalmarsson, and L. Ljung. Neural networks in system identification. *IFAC Proceedings Volumes*, 27(8):359–382, July 1, 1994. doi: 10.1016/S1474-6670(17)47737-8.

J. van Amerongen. Applications of adaptive control to ship steering. *IFAC Proceedings Volumes*, 18(5):1279 – 1284, 1985. ISSN 1474-6670. doi: 10.1016/S1474-6670(17)60740-7. 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation, York, UK, 3-7 July.

P. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, June 1967. doi: 10.1109/TAU.1967.1161901.

A Parameters for Simulation Study

The parameters used for the simulations in Chapter 5 are listed here. Table A.1 lists the parameters used in the prediction algorithm. Table A.2 lists the parameters used to create the disturbance. Table A.3 lists the parameters used for the forward backward filter. Table A.4 lists the hyperparameters of the RNN used in the prediction algorithm. The parameters of the nonlinear observer derived Fossen and Strand (1999) are given in Table A.5, the gains for the PID controller are listed in Table A.6, and the gains for the PI controller are presented in Table A.7.

Table A.1: Observer parameters

Parameter	1 DOF	3 DOF	Ship Model
N	1000	3000	3000
α	250	250	250
β	150	150	150
γ	180	180	180
ϵ	30	30	30

Table A.2: Disturbance parameters

Parameter	1 DOF	3 DOF	Ship Model
σ	20	20	-
ζ_{w}	0.01	0.01	-
Freq. range (rad/s)	[0.3 1.25]	[0.3 1.25]	0.9
$[K_{\mathrm{su}}\, K_{\mathrm{sw}}\, K_{\mathrm{y}}]$	-	[0.1 1 0.5]	-

Table A.3: Notch and Low-pass filter parameters for filters presented in Fossen (2011)

Parameter	Notch	Low-pass
Filter order	4	2
Design parameter ζ_{f}	0.05	$\sin(\pi/4)$
Notch frequency (rad/s)	w_0	-
Cut-off frequency (rad/s)	-	$w_0 + 0.2$

Table A.4: Hyperparameters of RNN

Hyperparameter	1 DOF	3 DOF	Ship Model
n_{u}	1	1	1
n_{y}	10	10	10
n_{fb}	6	6	2
Number of hidden layers	1	1	1
n_{h}	10	10	10
Hidden activation function	tansig	tansig	tansig
Preprocessing	Scaling	Scaling	Scaling
Training method	L-M	L-M	L-M
Regularization	$b = 10^{-8}$	$b = 10^{-8}$	$b = 10^{-8}$
	$a = 1 - b$	$a = 1 - b$	$a = 1 - b$
Training epochs	5	5	5
Loss function	MSEREG	MSEREG	MSEREG
Initial mu	0.001	0.001	0.001
Mu increase	10	10	10
Mu decrease	0.1	0.1	0.1
Maximum mu	10^{10}	10^{10}	10^{10}
Minimum gradient	10^{-10}	10^{-10}	10^{-10}

Table A.5: Tuning parameters of nonlinear observer derived in Fossen and Strand (1999)

Tuning parameter	Value
$\boldsymbol{\omega}_0$	$[0.9\,0.9\,0.9]^T$
$\boldsymbol{\zeta}$	$[0.1\,0.1\,0.1]^T$
$\boldsymbol{\zeta}_{\mathrm{n}}$	$[1.5\,1.5\,1.5]^T$
$\boldsymbol{\omega}_{\mathrm{c}}$	$[1.5\,1.5\,1.5]^T$
$\boldsymbol{\lambda}$	$\begin{bmatrix} 10^6 & 0 & 0 \\ 0 & 10^6 & 0 \\ 0 & 0 & 10^7 \end{bmatrix}$
$\boldsymbol{\kappa}$	$\begin{bmatrix} 10^7 & 0 & 0 \\ 0 & 10^7 & 0 \\ 0 & 0 & 10^8 \end{bmatrix}$
$\boldsymbol{M}$	$\begin{bmatrix} 2.47 \times 10^7 & 0 & 0 \\ 0 & 2.56 \times 10^7 & 7.41 \times 10^6 \\ 0 & 7.41 \times 10^6 & 4.64 \times 10^{10} \end{bmatrix}$
$\boldsymbol{D}$	$\begin{bmatrix} 4.97 \times 10^4 & 0 & 0 \\ 0 & 4.16 \times 10^6 & 0 \\ 0 & 0 & 9.75 \times 10^9 \end{bmatrix}$
$\boldsymbol{T}$	$\begin{bmatrix} 10^5 & 0 & 0 \\ 0 & 10^5 & 0 \\ 0 & 0 & 10^5 \end{bmatrix}$

Table A.6: PID-controller gains

Degree of freedom	K_P	K_I	K_D
Surge	1.76×10^{3}	8.97×10^{-3}	1.48×10^{2}
Sway	5.59×10^{-3}	9.05×10^{-3}	1.17×10^{1}
Yaw	3.53×10^{1}	8.77×10^{-3}	3.86

Table A.7: PI-controller gains

Tuning parameter	Value
$\boldsymbol{K}_P$	$\begin{bmatrix} 1.06 \times 10^{4} & 0 & 0 \\ 0 & 1.36 \times 10^{4} & -5.01 \times 10^{4} \\ 0 & 6.31 \times 10^{1} & 4.57 \times 10^{7} \end{bmatrix}$
$\boldsymbol{K}_I$	$\begin{bmatrix} 9.86 & 0 & 0 \\ 0 & 9.98 & -3.74 \times 10^{2} \\ 0 & 0.04 & 9.98 \times 10^{4} \end{bmatrix}$